LEARNING UNDER LIMITED SUPERVISION FOR BETTER GENERALIZATION

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Jae Myung Kim
aus Seoul / Südkorea

Tübingen
2025

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlinchen Fakultät der
Eberhard Karls Universität Tübingen

Tag der mündlichen Qualifikation:	25.06.2025
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatterin:	Prof. Dr. Zeynep Akata
2. Berichterstatter:	Prof. Dr. Seong Joon Oh

DEPARTMENT OF
COMPUTER SCIENCE

LEARNING UNDER LIMITED SUPERVISION FOR BETTER GENERALIZATION

JAE MYUNG KIM

Master of Science in Computer Science and Engineering

Adviser: Zeynep Akata
Full Professor, Technical University of Munich

Examination Committee

Chair: Gerard Pons-moll
Full Professor, University of Tübingen

Adviser: Zeynep Akata
Full Professor, Technical University of Munich

Members: Prof. Seong Joon Oh
Full Professor, University of Tübingen

Prof. Cordelia Schmid
Full Professor, Inria, Ecole normale supérieure, CNRS, PSL Research University

DOCTORATE IN COMPUTER SCIENCE

University of Tübingen, International Max Planck Research School for Intelligent Systems (IMPRS), and European Laboratory for Learning and Intelligent Systems (ELLIS)
April, 2025

Acknowledgements

Of all the gifts that this PhD journey has given me over the year, the greatest, without a doubt, has been the people I have met along the way. None of it would have been possible without the incredible support from those who stood by me, both academically and personally.

First and foremost, I am deeply grateful to my main advisor, Zeynep Akata, and my co-advisor through the ELLIS program, Cordelia Schmid, for their continuous support and invaluable guidance from the beginning till the end. Their mentorship has shaped me into becoming a researcher, from developing ideas to engaging in meaningful discussions and embedding the content into the writing. I am also grateful to Zeynep for her emotional and mental support, which helped me stay grounded and finish this long journey well. I also thank my TAC and thesis committee members, Seong Joon Oh, Wieland Brendel, and Gerard Pons-Moll, for their insightful feedback and encouragement throughout the process.

During my PhD, I was fortunate to be surrounded by a truly supportive group at EML, both inside and outside of research. I would like to begin by thanking Stephan and Sophia for their countless hours of thoughtful discussion and collaboration across every project we shared. My sincere thanks also go to Jessie, Karsten, and Nishad, whose perspectives inspired me during our time working together. Outside of academic life, I have unforgettable memories of travels with Thomas, Shyam, Massimo, Massi, and Quentin. Sharing office 129 in Munich with Leander and Yiran brought daily joy, running with Lukas in Tübingen relieved stress, and I am thankful to all other EML members who made each day brighter. A special thanks to Michael and Charlotte for their constant help with bureaucracy. Thanks to them, I could stay focused on my life and research while living in Europe.

Beyond EML, I was grateful to connect with wonderful people across different cities. During my first two years in Tübingen, I appreciate the friendship of Haiwen and Alex, and the regular tennis play with Auguste and Elif. I also thank Hyunjae, Seong Joon, and Seong Ah for their warm Korean community support. In Munich, I'm grateful to Sungho, Donghyun, and Wontaek for playing tennis regularly. During my time in Paris,

I'm thankful to Zerui and Ricardo for sharing the emotion of the final stretch toward graduation.

Finally, no words can truly express my gratitude to my parents, my brother, and my fiancée Hyojin for their unwavering support. Their love, patience, encouragement, and belief in me have made this journey possible. Their presence has been my strongest foundation.

I now consider Tübingen and Munich my second homes. All the big and small moments I experienced in Europe will stay with me forever, always bringing a smile whenever I look back. My PhD journey has not only shaped my research and career, but also given me unforgettable memories and lifelong friendships. To everyone who shared this chapter with me, thank you.

Abstract

Learning under limited supervision has been a long-standing theme in machine learning. This thesis explores how limited supervision can be effectively leveraged in the domains of computer vision and vision-language modeling. It covers a range of settings, from scenarios with no access to training data, to those with only a few labeled examples, and further to human intervention to AI systems that incorporate minimal but meaningful feedback. Across these settings, the central goal is to improve model generalization with limited real data.

We first consider scenarios where no training data is available for a downstream task in the vision-language models. We explore how to better utilize pre-trained foundation models without additional supervision. WaffleCLIP investigates the role of class descriptors in the zero-shot classification of the vision-language models and shows that performance gains stem from an ensemble of multiple noisy prompts rather than the semantic richness of the prompts, throwing a question of current prompting techniques. FLM introduces a feasibility prediction module that leverages large language models to assess the feasibility of novel state-object compositions for each class, thereby improving performance in open-world compositional zero-shot learning by discarding infeasible classes in the class candidate set. Together, these contributions improve the downstream performance of vision-language models in zero-shot settings.

Next, we examine the case where users have access to a small number of labeled examples. To maximize the utility of this data, we propose synthetic data generation frameworks guided by few-shot real examples where the synthetic data is used as training data for the downstream classification tasks. DataDream fine-tunes text-to-image generative models using LoRA on a handful of class-specific images, allowing the fine-tuned models to generate more realistic and discriminative data. Building on this, LoFT fine-tunes LoRA modules for each image, and then fuses two LoRAs within the same class at inference time, improving the fidelity and diversity of generated samples. These works demonstrate that synthetic data can provide complementary information to real data.

Finally, we explore the role of human intervention in interpretable models. While Concept Bottleneck Models allow humans to modify intermediate concept predictions,

current approaches require extensive manual correction. To address this, we propose a Concept Realignment module that reduces the number of necessary interventions by automatically adjusting related concepts after a small correction by human. This leads to more efficient human-in-the-loop collaboration, making interpretable models more practical in real-world applications.

Together, these contributions present a comprehensive study of learning under limited supervision in computer vision and multimodal tasks. We demonstrate that through thoughtful leverage of the foundation model, synthetic data augmentation, and efficient human interaction, it is possible to improve performance with minimal labeled data. This thesis lays the groundwork for future research toward data-efficient, adaptive, and human-aligned AI systems.

Zusammenfassung

Lernen unter begrenzter Aufsicht ist seit Langem ein zentrales Thema im Bereich des maschinellen Lernens. Diese Dissertation untersucht, wie begrenzte Aufsicht effektiv in den Bereichen der Computer Vision und des Vision-Language-Modellings genutzt werden kann. Dabei umfasst diese Dissertation eine Vielzahl von Szenarien – von Fällen ohne Zugang zu Trainingsdaten, über solche mit nur wenigen annotierten Beispielen bis hin zu menschlicher Intervention in KI-Systeme, die minimales, aber aussagekräftiges Feedback einbeziehen. In all diesen Szenarien ist das zentrale Ziel, die Generalisierungsfähigkeit von Modellen mit begrenzten realen Daten zu verbessern.

Zunächst betrachten wir Szenarien, in denen keine Trainingsdaten für eine nachgelagerte Aufgabe in Vision-Language-Modellen verfügbar sind. Wir untersuchen, wie vortrainierte Foundation-Modelle ohne zusätzliche Aufsicht besser genutzt werden können. WaffleCLIP analysiert die Rolle von Klassenbeschreibungen bei der Zero-Shot-Klassifikation von Vision-Language-Modellen und zeigt, dass Leistungsgewinne eher durch ein Ensemble mehrerer verrauschter Prompts als durch deren semantische Tiefe entstehen – was aktuelle Prompting-Techniken infrage stellt. FLM führt ein Modul zur Realismus-Vorhersage ein, das große Sprachmodelle nutzt, um den Realismus neuartiger Objekt-Zustands-Kompositionen je Klasse einzuschätzen. Dadurch wird die Genauigkeit im offenen, kompositionellen Zero-Shot-Lernen verbessert, indem unrealistische Klassenkandidaten verworfen werden. Diese Beiträge verbessern die Downstream-Performance von Vision-Language-Modellen in Zero-Shot-Szenarien.

Anschließend betrachten wir den Fall, in dem Nutzende Zugang zu einer kleinen Anzahl annotierter Beispiele haben. Um den Nutzen dieser Daten zu maximieren, schlagen wir Frameworks zur Generierung synthetischer Daten vor, die durch wenige reale Beispiele geleitet werden, wobei die synthetischen Daten als Trainingsdaten für nachgelagerte Klassifikationsaufgaben dienen. DataDream passt text-zu-Bild-Generierungsmodelle mithilfe von LoRA auf eine Handvoll klassenspezifischer Bilder an, sodass die angepassten Modelle realistischere und differenziertere Daten erzeugen können. Darauf aufbauend passt LoFT LoRA-Module für jedes Bild einzeln an und fusioniert dann für die Inferenz zwei LoRA-Module derselben Klasse, was die Qualität und Vielfalt der generierten Beispiele

erhöht. Diese Arbeiten zeigen, dass synthetische Daten reale Daten sinnvoll ergänzen können.

Abschließend untersuchen wir die Rolle menschlicher Intervention in interpretierbaren Modellen. Während Concept Bottleneck Models es Menschen ermöglichen, Zwischenvorhersagen von Konzepten zu modifizieren, erfordern aktuelle Ansätze umfangreiche manuelle Korrekturen. Um dem entgegenzuwirken, schlagen wir ein Concept Realignment-Modul vor, das die Anzahl notwendiger Eingriffe reduziert, indem es nach einer menschlichen Korrektur automatisch verwandte Konzepte anpasst. Dies führt zu effizienterer Human-in-the-Loop-Zusammenarbeit und macht interpretierbare Modelle praktikabler für reale Anwendungen.

Diese Beiträge zusammen liefern eine umfassende Untersuchung des Lernens unter begrenzter Aufsicht in Computer-Vision und multimodalen Aufgaben. Wir zeigen, dass durch den durchdachten Einsatz von Foundation-Modellen, synthetischer Datenaugmentation und effizienter menschlicher Interaktion die Leistung mit minimal gelabelten Daten verbessert werden kann. Diese Dissertation bildet die Grundlage für zukünftige Forschung hin zu daten-effizienten, adaptiven und menschzentrierten KI-Systemen.

Contents

List of Figures

List of Tables

Listings

1

Introduction

Nature has long demonstrated remarkable efficiency in learning from limited examples. A child can recognize cats after seeing just a few specimens, and can learn the language by capturing auditory features from adult speech. This ability to generalize from sparse data represents one of the most fundamental aspects of natural intelligence. Machine learning systems, by contrast, have historically required vast quantities of labeled data to achieve comparable generalization. This disparity between natural and artificial learning efficiency has motivated a long-standing research agenda which is to build AI systems that learn effectively from limited supervision.

In machine learning, numerous approaches have been proposed to train the model from limited supervision. Early work in Bayesian learning, semi-supervised learning, weakly-supervised learning, transfer learning, few-shot learning, or meta-learning explored how the model could benefit from minimal supervision. Most recently, the advent of foundation models has reshaped the landscape. These models offer impressive generalization capabilities, yet their performance in domains with limited labeled data remains an open and active area of research.

In this thesis, we visit the problem of limited supervision in the context of modern foundation models, particularly within computer vision and vision-language tasks. We examine three key directions: (1) how to better leverage foundation models in zero-shot setting to improve the performance of the downstream tasks, (2) synthetic data generation for downstream model training in few-shot scenarios, and (3) how to build interpretable models that enable efficient human intervention. Each of these directions addresses a distinct facet, contributing toward more generalized AI systems. The remainder of this chapter first provides a historical overview of learning under limited supervision, tracing its evolution from early paradigms to current practices. We then introduce key challenges in vision-language models, synthetic data generation, and human intervention to the AI models. Finally, we summarize the core contributions of this thesis and outline how each chapter builds upon the broader goal of learning with less.

1.1 Historical Background

The challenge of learning under limited supervision has been a long-standing theme in machine learning. From the earliest days of the field, researchers have recognized that labeled data is often expensive, noisy, or scarce. Before the deep learning era, the lack of large-scale labeled data naturally led researchers to explore how models could learn from limited supervision. This practical constraint drove developments in statistical learning approaches that could generalize effectively from small datasets [16, 204]. Techniques such as kernel methods [176, 179] and Bayesian approaches [16, 198] provided theoretical frameworks for handling limited data scenarios. Also, there was a development of approaches such as semi-supervised learning [138, 245], self-taught learning [156], and weak supervision [38, 41] where models were trained using incomplete or indirect labels. Early work on active learning also addressed this challenge by selecting the most informative examples for labeling [177].

With the rise of deep learning in the 2010s, large-scale datasets like ImageNet [42] enabled unprecedented performance in tasks such as image classification and object detection. However, these advances came with a growing dependence on large labeled datasets, which limited applicability in domains where such data is difficult to obtain. This led to growing interest in transfer learning, where models pre-trained on large-scale datasets are fine-tuned on smaller, task-specific datasets [214, 223]. Data augmentation techniques [39, 43, 186, 231] also emerged as crucial tools to artificially expand limited training data.

Around the same time, few-shot learning emerged as a separate line of research, aiming to recognize new classes from just a few labeled examples [190, 205, 210]. For instance, Prototypical Networks [190] reformulated classification as a metric learning problem, where models learn to compare query images with support examples rather than directly classifying them. To support such generalization, meta-learning, also known as learning to learn, has been explored where models are trained to adapt quickly to new tasks using limited supervision [57, 79, 136]. For instance, the Model-Agnostic Meta-Learning [57] method optimizes for initial parameters that can be efficiently fine-tuned with just a few gradient steps on novel tasks.

Another related direction that gained traction was zero-shot learning (ZSL), which aimed to recognize unseen classes without any labeled examples during training. Classical ZSL methods relied on semantic embeddings such as attribute vectors or word embeddings to relate seen and unseen categories [1, 58, 160, 217]. For instance, embedding-based methods like DeViSE [58] aligned visual and word embeddings to transfer knowledge to novel categories.

The emergence of foundation models [19, 110, 155] in the 2020s has fundamentally changed how zero-shot and few-shot learning are understood and applied. These models

offer powerful pre-trained architectures that are trained on millions to trillions of data points. Unlike traditional models that were trained for specific tasks, these foundation models serve as general-purpose systems that can be adapted to numerous downstream applications with minimal task-specific data or fine-tuning. In this context, zero-shot and few-shot learning typically refer to directly applying a pre-trained model to a new task. For instance, in the case of few-shot learning, it is often possible that the relevant concepts in the downstream data were partially observed or learned by the model during the pre-training phase, which distinguishes it from the classical few-shot setting that assumes completely novel classes or tasks. This distinction has prompted researchers to reconsider the terminology of few-shot learning, with some suggesting that foundation models are performing a form of transfer learning rather than few-shot learning in the classical sense. Alongside these conceptual arguments, foundation models have dramatically advanced the state-of-the-art in solving tasks with limited supervision across domains.

In vision-language models (VLMs) [30, 84, 106, 107, 110, 155], zero-shot learning is performed by leveraging textual prompts, either by class names or natural language descriptions, to guide the model in classifying or retrieving images [141, 152, 155]. This allows the model to perform tasks like image classification without having seen any labeled examples of the target classes. The effectiveness of these approaches has been further improved through prompt ensembling techniques [128, 162].

In few-shot settings, when a small number of real images from the target task are available, they can be used to either fine-tune the pre-trained model [9, 61, 194, 243] or to serve as reference examples in a retrieval-style setup [237, 238] where the model compares a query image to the provided support images. For instance, Parameter-efficient fine-tuning methods like adapters [61] and prompt tuning [9, 243] have emerged as effective approaches to adapt foundation models to downstream tasks with limited data.

In large language models [5, 6, 19, 32, 143, 199, 200], few-shot learning often relies on in-context learning [19, 44], where a few examples are included directly in the input prompt at inference time. This enables the model to adapt to a new task without parameter updates, by mimicking the desired behavior based on the provided examples. Moreover, various techniques have been explored to optimize in-context learning, including example selection strategies [111, 166], chain-of-thought prompting [98, 212], and retrieval-augmented generation [83, 104].

Despite their broad capabilities, foundation models still face challenges in compositional ability, fine-grained discrimination, and visual/semantic understanding [114, 175, 195, 229]. These factors limit the effectiveness of foundation models in zero-shot and few-shot learning scenarios. In zero-shot cases, performance can be highly sensitive to prompt formulation, and models may struggle to distinguish between visually similar classes (e.g., different bird species with subtle differences in feather patterns) or polysemy words (e.g., clothing "iron" vs. material "iron"). In few-shot settings, models often require carefully selected or balanced examples (e.g., ensuring that each class in a rare disease classification task is represented with similar image quality or diversity), and may overfit

to spurious correlations in small datasets (e.g., associating background color with object category due to a biased training subset), limiting their ability to generalize to new inputs.

Beyond passively consuming labeled data, another increasingly important direction involves human interaction with AI models, particularly in few-shot scenarios [97, 189]. In real-world applications, few-shot supervision sometimes arises not from curated datasets but through human-in-the-loop interactions, where users provide feedback, corrections, or examples during deployment. This type of supervision is inherently limited in quantity, making the efficacy of human intervention critical. For AI systems to serve as effective assistants, they must be designed to not only perform well with limited examples, but also to understand and respond meaningfully to human input. This includes being sensitive to user feedback, incorporating corrections into future predictions, and maintaining transparency in their decision-making process.

Belonging to the said field of research, this thesis aims to explore how AI models can generalize effectively under limited supervision and how human can effectively intervene the interpretable model to improve the model performance.

1.2 Contributions

This thesis primarily explores learning under limited supervision in the computer vision domain. It addresses challenges in zero-shot generalization, data augmentation under few-shot guidance, and human intervention in the AI system.

The thesis makes three key contributions. First, it aims to improve how models operate in data-scarce environments by leveraging pre-trained foundation models. Second, it introduces methods for synthetic data generation guided by few-shot real images to create high-quality training data that improves model performance. Third, it proposes a framework for efficient human intervention in interpretable AI models, allowing users to correct model decisions with minimal trial to achieve the target performance.

The following sections describe these contributions in detail.

1.2.1 Zero-Shot Learning in Vision-Language Models

Zero-shot learning is a paradigm where a model is expected to classify new, unseen categories without explicit training examples. With the rise of foundation models, the definition of zero-shot learning has expanded beyond its traditional form. In these large-scale pre-trained models, zero-shot learning often refers to the ability to perform downstream tasks directly using the pre-trained model, without fine-tuning on task-specific data. In vision-language models, this is basically done by giving textual prompts with the class names of downstream benchmarks. While this approach has demonstrated

impressive capabilities, its effectiveness remains inconsistent since the alignment between textual descriptions and visual features is often weak.

We investigate how foundation models can be leveraged in zero-shot learning scenarios of the vision-language models (VLMs). VLMs classify images based on textual descriptions, yet the way these descriptions influence classification performance is not well understood. The first contribution examines whether language-model-generated descriptions genuinely enhance classification or if their performance gains stem from an ensemble of multiple prompts. Our findings reveal that noisy prompts can achieve similar improvements to semantic descriptions, and further performance gains can be obtained by strategically combining both. This warns us to be cautious about interpreting the model's improvement, and systematic study is needed to understand the behavior of VLMs.

The second contribution is in the field of open-world compositional zero-shot learning where each class is composed of state and object pair. We introduce a feasibility prediction pipeline that leverages large language models to assess the feasibility of unseen state-object compositions, helping VLMs make more reliable and semantically coherent predictions by removing unfeasible classes from the class set for prediction.

1.2.2 Synthetic Data Generation with Few-shot Guidance for Data Augmentation

Synthetic data generation is a technique where a generative model is used to create synthetic data, which is then employed as a training set for a downstream task. This approach is particularly valuable when the availability of real training data is limited, expensive, or difficult to collect. By augmenting small datasets with high-quality synthetic samples, the downstream models can generalize better, reducing the reliance on large-scale manual annotation efforts.

This thesis investigates how text-to-image generation models, specifically Stable Diffusion [159], can be leveraged to generate synthetic training data for the downstream classification tasks. A naive approach to generating synthetic data is to use a class name as a text prompt and pass it through a pre-trained diffusion model to produce class-representative images. However, this method suffers from several key limitations. First, semantic ambiguity in language models can lead to misinterpretations of prompts, generating objects that do not match the intended category (e.g., generating an image of the material "iron" instead of the clothing "iron"). Second, pre-trained diffusion models struggle with fine-grained visual details, making them unreliable for tasks requiring precise visual differentiation.

To address these challenges, this thesis explores how fine-tuning generative models on a few real images can significantly improve synthetic data quality and downstream classification performance. The key insight is that pre-trained diffusion models lack sufficient domain-specific alignment, which leads to inconsistent object representations. By introducing few-shot fine-tuning techniques, we ensure that generated images better

reflect real-world distributions and maintain the granularity needed for fine-grained classification tasks. This approach improves the realism, diversity, and relevance of synthetic data, making it a more effective augmentation strategy for training classification models in low-data regimes. Beyond fine-tuning, this work also examines how fusing multiple fine-tuned models can further enhance data diversity without sacrificing fidelity. This fusion approach further improves the generalization of the classification model.

1.2.3 Effective Human Intervention for Interpretable AI

An interpretable model is an AI system whose decision-making process can be understood by human users. Unlike traditional deep learning models, which operate as black boxes, interpretable models provide reasoning behind their predictions, allowing users to assess and trust the model's decisions. Beyond transparency, a key advantage of interpretable models is that they enable human interaction, allowing users to intervene and refine the model's predictions. This ability to directly modify the model's reasoning process is particularly valuable in high-stakes applications where errors can have significant consequences.

This thesis investigates how Concept Bottleneck Models (CBMs) can facilitate human intervention to improve model performance. CBMs first predict human-understandable concepts (e.g., "has a spotted pattern" or "has a black wing" for bird classification) before making a final classification decision. This structured representation allows users to inspect which concepts contributed to the model's decision, making AI predictions interpretable. Furthermore, users can directly modify incorrect concept predictions, which in turn adjusts the final classification output. However, despite the advantages of CBMs in human-AI collaboration, their effectiveness is limited by the efficiency of user intervention. If corrections require too many manual edits, the practical benefits of interpretability diminish, making CBMs difficult to scale in real-world applications.

To overcome this limitation, this thesis introduces a concept intervention realignment mechanism that improves the efficacy of human intervention in CBMs. Instead of requiring manual corrections for every mispredicted concept, our approach enables the model to automatically adjust related concepts once a small number of corrections have been made. By leveraging concept dependencies and structured realignment, our method reduces the burden on human users, making interventions more scalable and efficient. This structured correction framework ensures that minimal human effort leads to maximum improvement in model performance.

1.3 Outline

This section presents a concise summary of each chapter in this thesis, highlighting relevant publications and collaborations, along with their contributions to the overall research. The content covered in these chapters is based on published work. The overview of the overall

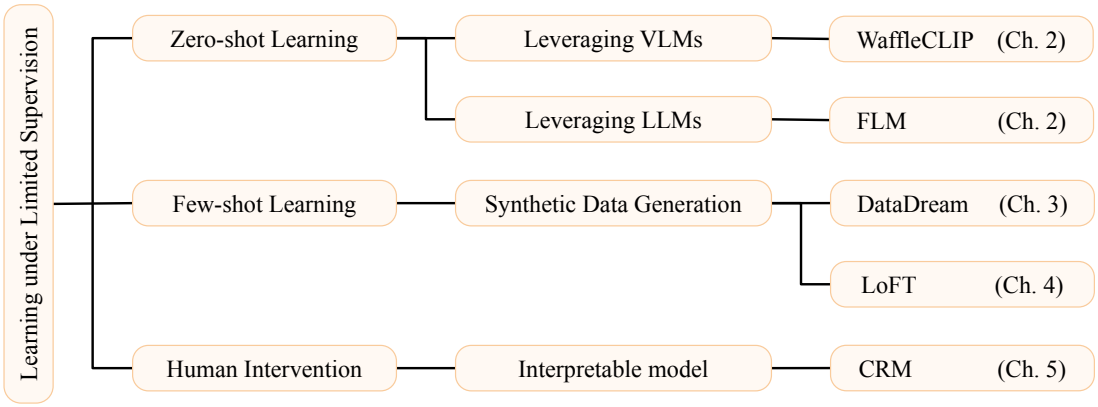


Figure 1.1: The overview of the main theme and categorization of the thesis.

theme and categorization of each work is shown in Figure 1.1. Appendix E provides a comprehensive list of publications, their respective co-authors, and the contribution of each author for each publication.

Chapter 1: Introduction introduces the historical background and outlines the challenges of learning under limited supervision in computer vision. The background starts from the pre-deep learning era, deep learning in 2010s, to the era of foundation models, and provides the list of methods suggested in each era to train AI models under limited supervision. The chapter then defines the primary research directions explored in this thesis: (1) leveraging foundation models for zero-shot learning, (2) improving model generalization through synthetic data generation, and (3) enabling minimal human intervention to refine interpretable models. Lastly, this chapter contains a structural overview of the thesis.

Chapter 2: Zero-shot Learning by Leveraging Foundation Models introduces two projects that are leveraging foundation models to improve the downstream performance of the vision-language models in the zero-shot setting. We first introduce our WaffleCLIP method which studies and leverages the inherent ability of the vision-language models. We question the performance improvement when leveraging language-model-generated textual descriptions, whether the improvement attributes from meaningful semantic enrichment or merely structured noise effects. The study finds that structured noise alone can achieve comparable performance to semantic descriptors and proposes a strategy to combine both the noise and descriptors for further gains. This work was published at ICCV 23 as a shared first-author paper together with Karten Roth and a collaboration with Sophia Koepke.

Second, we introduce our FLM method which predicts the feasibility of each class for open-world compositional zero-shot learning. For feasibility prediction, FLM leverages large language models to better comprehend the semantic relationships between states and objects of the given class. By identifying feasible state-object combinations, the proposed approach prunes the class list for prediction, reducing the number of implausible

candidates in open-world scenarios where all possible state-object combinations are considered as candidate classes. This improves the accuracy and reliability of zero-shot classification by focusing on realistically co-occurring compositions. This work was published at ECCV 2024 OODCV Workshop. Jae Myung Kim participated in this project as a first author, and it was done in collaboration with Stephan Alaniz.

Chapter 3 DataDream: Few-shot Guided Dataset Generation introduces a method for synthetic data generation which is used as a training dataset for classification tasks. Motivated by the observation of the text-to-image generative models that the standard prompts or even rich sentences sometimes do not generate the object of interest, we propose that few-shot guidance is necessary to achieve faithful image generation. We propose DataDream, a method that fine-tunes LoRA weights of the generation model on a few real images, and generates training data using the adapted model. This improves the fidelity of synthetic images, leading to the improved performance of the classification models. This work was published at ECCV 2024 as a shared first-author paper with Jessica Bader and a collaboration with Stephan Alaniz.

Chapter 4 LoFT: LoRA-Fused Training Dataset Generation with Few-shot Guidance introduces our LoFT method, a follow-up work of the DataDream project. Motivated by the observation that the DataDream is sometimes unstable in training generative models with few-shot images, we propose LoFT that has more stability in training by fine-tuning LoRA weights on individual real images. To achieve diversity of the generated images, after training is done, LoFT fuses random two LoRA weights within the same class at inference time, producing synthetic images that combine the features of corresponding real images. This work is under submission. Jae Myung Kim participated in this project as a first author, and it was done in collaboration with Stephan Alaniz.

Chapter 5 Improving Intervention Efficacy via Concept Realignment in Concept Bottleneck Models introduces the module that improves the efficacy of human intervention in the AI model. We propose CRM, a concept intervention realignment module that learns statistical concept relations. It realigns the concept values after an intervention has been performed in the concept bottleneck models. This reduces the number of interventions needed to achieve the target performance. This work was published at ECCV 2024. Jae Myung Kim participated in this project as a second author, in collaboration with Nishad Singhi and Karsten Roth.

Chapter 6 Discussion and Conclusion reflects on the thesis's contributions and their broader impact. It discusses the key findings and how they advance research in the limitation of data. Additionally, it outlines the future research direction.

2

Zero-shot Learning by Leveraging Foundation Models

In this chapter, we present two lines of work that do not use any of the training data but rather leverage the foundation models to improve the downstream performance of the vision-language models in a zero-shot way.

In Section A.1, we question the true source of performance gains in descriptor-based prompts for zero-shot classification of vision-language models. We show that improvements often stem from the structure of averaging multiple prompts rather than the semantic content of the descriptors themselves. We propose WaffleCLIP, a method that uses multiple noisy prompts to achieve similar classification performance compared to the descriptor-based prompts. We further propose to use both noise and descriptor in the prompts to enable better classification results.

In Section A.2, we present FLM which addresses the problem of open-world compositional zero-shot learning. This method leverages large language models to predict the feasibility of unseen compositions between a state and object for a given class name. This allows to filter out classes with infeasible combinations, reducing the number of class candidates when predicting the class of the image input. This leads to an improved performance of the open-world compositional zero-shot classification.

2.1 Waffling around for Performance: Visual Classification with Random Words and Broad Concepts

The visual classification performance of vision-language models such as CLIP has been shown to benefit from additional semantic knowledge from large language models (LLMs) such as GPT-3. In particular, averaging over LLM-generated class descriptors, e.g. "waffle, *which has a round shape*", can notably improve generalization performance. In this work, we critically study this behavior and propose `WaffleCLIP`, a framework for zero-shot visual classification which simply replaces LLM-generated descriptors with random character and word descriptors. **Without** querying external models, we achieve comparable performance gains on a large number of visual classification tasks. This allows `WaffleCLIP` to both serve as a low-cost alternative, as well as a sanity check for any future LLM-based vision-language model extensions. We conduct an extensive experimental study on the impact and shortcomings of additional semantics introduced with LLM-generated descriptors, and showcase how - if available - semantic context is better leveraged by querying LLMs for high-level concepts, which we show can be done to jointly resolve potential class name ambiguities. Code is available here: https://github.com/ExplainableML/WaffleCLIP.

2.1.1 Introduction

Task-specific tuning of natural language prompts [27, 81, 112, 243] can improve the performance of large vision-language models (VLMs) [155]. However, if the model does not have access to additional training data, i.e. in the zero-shot setting, this is not an option. Instead, a promising alternative [128, 141, 152] is querying large language models (LLMs) to provide additional semantic context to enrich class representations. Extending classnames with fine-grained class descriptors generated by GPT-3 [19] and minimal human intervention has experimentally shown to boost results [128, 152], for instance with class-based descriptors on top of classnames, e.g. *a round shape* for *waffle* [128].

However, close inspection of GPT-3 generated semantic cues indicates a high degree of diversity, limited visual relevance, and ambiguity [128]. For instance, multiple descriptors can be assigned to the same class despite likely not co-occurring, e.g. *"steamed"* and *"fried"*, or non-visual attributes might be mentioned, e.g. *"a sour and spicy smell"*, or the class interpretation might be ambiguous, e.g. *"webbed feet"* for *"Peking duck"* as a food item. Hence, the underlying drivers of performance improvements when using generated fine-grained class descriptors are unclear.

To understand these performance gains, we first show that each set of class-specific GPT-3 generated descriptors can be replaced with a fixed set of randomly selected, class-independent descriptors while still retaining similar benefits in performance. Motivated by this observation, we take this one step further and propose `WaffleCLIP`, named after *waffling around* the class name, that replaces the LLM-generated fine-grained descriptors,

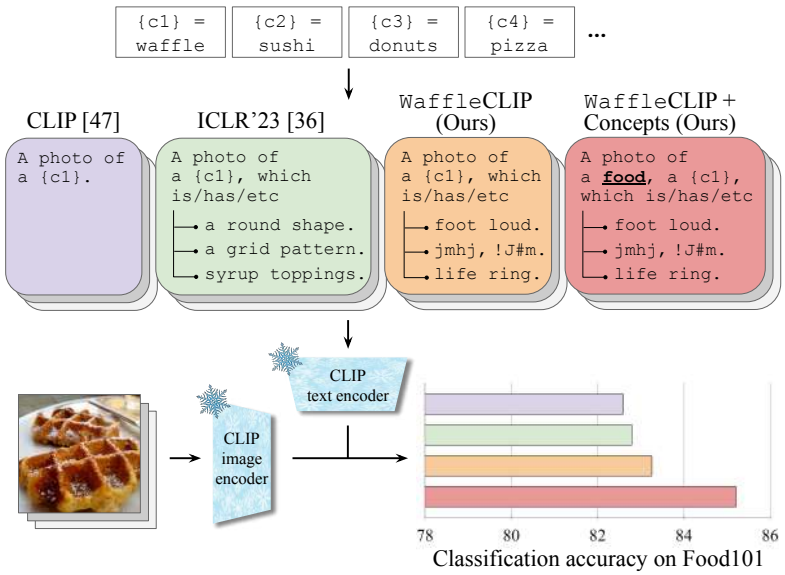


Figure 2.1: Substituting GPT-3 generated fine-grained descriptors with random word or character sequences yields competitive performance. High-level concepts further resolve classname ambiguities for additional gains.

e.g. *a round shape, a grid pattern*, with random words (e.g. *"foot loud"*) or character lists (e.g. *"jmhj, !J#m"*) based on average class name length and word counts (cf. Figure 2.1). As WaffleCLIP does not require access to LLMs for additional context (unlike e.g. [128, 141, 152, 201]), it remains *inherently zero-shot*. Consequently, it also serves as an important sanity check for future methods utilizing external model queries.

Naturally, the convincing performance of WaffleCLIP across benchmarks raises questions regarding the true benefits of additional semantics introduced by LLM-generated descriptors. We provide answers with extensive experiments, showcasing that semantic descriptors produced by LLMs offer a *structurally* different and *complementary* impact on the classification behavior. However, we find this not to be fully driven by additionally introduced semantics, but rather a different form of structured noise ensembling. Instead, we show that actual semantic context is better introduced through coarse-grained, high-level concepts. Given access to external LLMs, we suggest a query mechanism for GPT-3 to automatically generate these concepts (e.g. *food* for *waffle, peking duck*), while jointly resolving issues of context-dependent class label ambiguity.

In summary, our contributions are: **1)** We motivate and propose WaffleCLIP to use random character and word descriptors to enhance the semantic retrieval process in VLMs (particularly CLIP); **2)** we demonstrate that WaffleCLIP yields comparable zero-shot classification performances at lower cost compared to methods reliant on LLM-generated descriptors, thus also serving as an important sanity check for future models; **3)** we extensively study the semantic context introduced through LLM-generated descriptors and propose (automatically extracted) high-level LLM-generated concepts as an alternative for better use of semantics while tackling classname ambiguities.

2.1.2 Related Work

Image classification with VLMs such as CLIP [155] has gained popularity particularly in low-data regimes. As input prompts have a significant impact on the performance, recent research has focused on the exploration of learnable prompts for the text encoder [119, 187, 242, 243], the visual encoder [9, 26, 117, 216] or for both encoders jointly [219]. Alternatively, synthetic images generated from the classnames can support image classification [**bansal2023leaving**, 69, 201]. In contrast, we do not tune prompts or query image generation methods, but propose to use prompts containing random characters or words to enhance the zero-shot capabilities of VLMs.

Adding external knowledge to language prompts. Recently, multiple works have leveraged LLMs to obtain more effective prompts. [124, 131, 152] utilized GPT-3 [19] to produce and study lengthy, descriptive sentences that articulate the visual concepts for each category, while [141] generated semantic hierarchies to identify subclasses of categories for zero-shot class prediction. [128] used multiple fine-grained LLM-generated class descriptors, which enhance accuracy and appear to provide interpretability by assigning weights to each descriptor. Similarly, different kinds of descriptions have been used for image classification, by manually crafting descriptions [70, 158], or by utilizing external databases based on Wikipedia [34, 52, 133, 147], the WordNet hierarchy [129, 164, 180], or the ImageNet-Wiki [21]. Whilst external knowledge from LLMs can be valuable, we can match the image classification performance of using fine-grained LLM-generated descriptors with randomly sampled characters and words as class descriptors. In addition, we find that if semantic context is available through LLMs, it is better integrated through high-level context (c.f. also [48]), for which we provide an automatic extraction mechanism.

Noise augmentation. Data augmentation through noise is known to enhance the performance and robustness of model training for a variety of tasks and domains [56, 186]. In the language domain, noise can be incorporated in the embedding or input space. For instance, [28, 66, 192] used linguistic embedding space augmentations inspired by mixup [235], and [31] added Gaussian embedding space noise. Augmentation through input space noise has been performed at the word- [96, 213], token- [213] or character-level [11, 71, 140, 171]. For character-level noise augmentation, characters are randomly substituted, added, or removed [11, 71, 140]. In all cases, these augmentation are used to prevent overfitting *during training*. Instead, our approach utilizes character- and word-level language augmentation to perturb the class prompts for improved zero-shot image classification.

2.1.3 Method

We first describe image classification using class descriptors following [128] (§2.1.3.1), before motivating and explaining our LLM-free, random semantic descriptor alternative `WaffleCLIP` (§2.1.3.2). Finally, if LLMs are available, we highlight a simple extension to

ViT-B/32	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	Avg
CLIP [155]	54.71	62.01	51.28	40.78	39.12	82.59	85.06	43.18	57.34
DCLIP [128]	**55.82**	**63.12**	52.47	**43.29**	40.47	82.79	86.54	**43.99**	**58.56**
DCLIP (same, 1x)	55.47 ±0.24	62.89 ±0.19	52.64 ±0.28	39.74 ±2.69	40.29 ±0.47	83.82 ±0.48	87.04 ±0.27	43.35 ±0.41	58.16 ±1.01
DCLIP (same, 2x)	55.75 ±0.21	63.10 ±0.19	**52.72** ±0.23	39.73 ±1.66	**40.61** ±0.22	**84.01** ±0.23	**87.10** ±0.14	43.29 ±0.22	58.29 ±0.62

Table 2.1: **Motivating random class descriptors.** Comparing CLIP [155] and the GPT-descriptor-extended CLIP [128] (DCLIP) with the same set of randomly sampled descriptors for each class, where the set size is either the average number of descriptors per class in DCLIP (*same, 1x*), or twice that (*same, 2x*). A random set of descriptors per class can match or even outperform DCLIP across backbone architectures (results for ViT-L/14 and ResNet50 are included in the suppl. material) confirming that randomized prompt averaging leads to higher performance.

incorporate semantics while jointly resolving ambiguities with automatically extracted high-level semantic concepts (§**??**).

2.1.3.1 Image classification with class descriptors

Given target categories C and a query image x, the zero-shot image classification protocol used in CLIP [155] defines the classification problem as nearest neighbour retrieval:

$$\tilde{c} = \arg\max_{c \in C} s(\phi_I(x), \phi_L(f(c))), \tag{2.1}$$

with prompt $f(c) =$ `"A photo of a {c}."` and image and language encoder ϕ_I and ϕ_L. To improve the retrieval process, [128] converts the simple class-embedding retrieval to a dictionary-based one, where a class c is associated with a set of descriptors D_c: `"{c} which (is/has/etc) {descriptor}."` with e.g. `c = "waffle"` and `descriptor = "a round shape"`. Given D_c for classes c, classification is reformulated as

$$\arg\max_{c \in C} \frac{1}{|D_c|} \sum_{d \in D_c} s(\phi_I(x), \phi_L(d)), \tag{2.2}$$

which defines the similarity between the image x and class c as the average similarity to all its descriptor variants. We abbreviate this descriptor-based extension of CLIP as *DCLIP*.

2.1.3.2 `Waffle`CLIP

DCLIP [128] [1] requires external LLMs for descriptors that convert the single-class matching problem to one over an ensemble of fine-grained class representations.

Motivation. We observe that LLM-generated class descriptors reveal high diversity, limited visual relevance, and ambiguity. From a conceptual perspective, this makes it hard to pin down the precise benefits of generated class descriptors used e.g. in [152] or [128]. To understand a possible driver of performance improvements, we conduct a simple experimental study, shown in Tab. 2.1. We take all available LLM-generated descriptors

[1] DCLIP [128] reports improvements over CLIP by using the phrase `"{c}, which (is/has/etc) {descriptor}."` instead of `"A photo of {c}, which (is/has/etc) {descriptor}."` as suggested in the original CLIP paper. For fair comparison with CLIP, we utilize the latter.

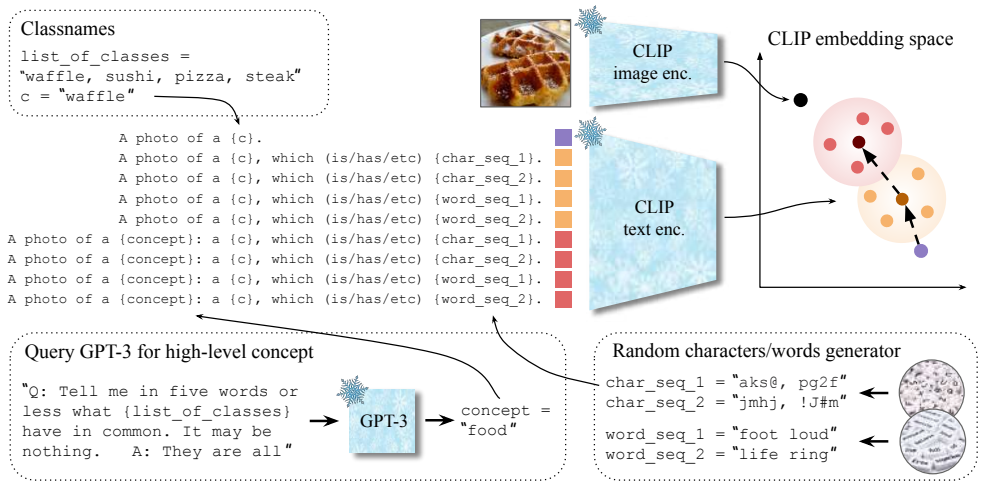


Figure 2.2: **Visual classification with WaffleCLIP using random characters/words.** Introducing character-level or word-level noise following the classname increases the similarity between the image features and text features (orange). WaffleCLIP can be further enhanced by adding a high-level concept descriptor in the prompt (red).

for a dataset from [128], sample a small set of descriptors where the cardinality of the set is the average number of descriptors per class used in DCLIP, and assign this same set of random descriptors to every class, i.e. *DCLIP (same, 1x)*. This shows a close match to DCLIP (e.g. 58.56% and 58.16% for ViT-B/32 in total average) and in parts even better performance (e.g. 0.83% improvement in Food101 for ViT-B/32). This reveals averaging over descriptor variations as one of the key drivers for performance. The results further improve when increasing the number of random LLM-generated descriptors for each class (*DCLIP (same, 2x)*, e.g. 58.16% $\rightarrow$ 58.29% for ViT-B/32). This indicates that the role of additional descriptor semantics is likely overestimated, especially when uncurated descriptors are used. Building on the benefits of averaging over various prompt variants to extract a better semantic representation estimate of an associated class, we investigate whether fully randomized prompt descriptors can provide similar benefits, **without** querying external LLMs.

WaffleCLIP. This motivates WaffleCLIP, an *LLM-free* descriptor alternative that uses simple randomized descriptors. In particular, we populate D_c with class-independent, random word sequences or random character lists, with a fixed number of characters per word l_w, and a fixed number of words n_w. For example, $l_w = 4$ and $n_w = 2$ for `char_seq_1 = "aks@, pg2f"` in Fig. 2.2. To avoid introducing hyperparameters, we leverage a simple heuristic where the average number of words and average number of characters per word in the provided class labels determines l_w and n_w. As a result, this converts the standard CLIP input prompt `"A photo of a {c}."` into `"A photo of a {c}, which (is/has/etc) {random_sequence}."`, where we follow the extension structure used in [128].

ViT-B/32	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	Avg
CLIP [155]	54.71	62.01	51.28	40.78	39.12	82.59	85.06	43.18	57.34
+ Concepts	↓	↓	52.23	48.86	39.31	84.66	86.73	↓	58.96
DCLIP [128]	55.82	63.12	52.47	43.29	40.47	82.79	86.54	**43.99**	58.56
WaffleCLIP (ours)	**55.92** ±0.08	**63.31** ±0.09	**52.38** ±0.12	44.31 ±1.07	40.56 ±0.07	83.25 ±0.21	85.70 ±0.25	43.16 ±0.25	58.57 ±0.41
+ Concepts	↓	↓	52.83 ±0.19	48.51 ±0.70	40.97 ±0.08	**85.21** ±0.06	87.52 ±0.10	↓	59.47 ±0.42
+ GPT descr. + Concepts	↓	↓	**52.77** ±0.26	**51.64** ±0.25	**41.35** ±0.09	84.87 ±0.05	**87.71** ±0.18	↓	**60.21** ±0.20

Table 2.2: **Image classification with WaffleCLIP** which extends input prompts with random word and character sequences and matches the performance of DCLIP [128] using GPT-generated class descriptors. Additional semantic context through high-level concepts (+ *Concepts*) can offer further boosts, particularly on benchmarks where classnames can be generic or ambiguous. We further find that WaffleCLIP complements the use of GPT-generated descriptors (+ *GPT descr.*). (↓) denotes same results as previous lines where high-level concept guidance is not applicable. For ViT-L/14 and RN50, see Supp.

2.1.3.3 Better semantics and reduced ambiguity via high-level concepts

Due to the limited impact of additional semantics introduced by fine-grained descriptors (c.f. §2.1.3.2), we propose an alternative way of querying LLMs that does not require averaging across multiple descriptors and simultaneously addresses the issue of class ambiguities. Therefore, we suggest taking a step back and searching not for additional class details, but instead for higher-level commonalities *between* the classes, akin to the use of class hierarchies in image classification [64]. Understanding commonalities between multiple target classes can help resolve ambiguities. If the class `"boxer"` is seen in the context of animal classification, it likely refers to the animal instead of a human athlete. We propose to automatically produce such high-level concepts by using available class names (or subsets if the class count exceeds the maximum LLM input sequence length) $C_{\mathcal{D}}$ for a dataset $\mathcal{D}$ and querying GPT-3 [19] with:

```
"Q: Tell me in five words or less what {list_of_classes} have in common.
                    It may be nothing. A: They are all ".
```

After extracting the shared `concept`, simple filtering of concepts is executed to check if generated concepts are non-specific, namely `"Object"`, `"Thing"`, `"Verb"`, `"Adjective"`, `"Noun"`, or `"Word"`. If so, high-level concept guidance is omitted (this is only the case for three out of eleven visual classification benchmarks, see also §**??**). We then augment the default CLIP prompt to `"A photo of a {concept}: a {c}."` and for WaffleCLIP, the prompt is extended to `"A photo of a {concept}: a {c}, which (is/has/etc) {random_sequence}."` While the prompt style can likely be improved, this naive extension already offers remarkable benefits.

2.1.4 Experiments

We first provide implementation details before comparing WaffleCLIP to DCLIP in §2.1.4.1. Extending our observations in Tab. 2.1, we study the source of performance gains via LLM-generated descriptors (§2.1.4.2) and present a better way for introducing semantics into the retrieval process while tackling semantic ambiguities with automatically extracted

high-level concepts (§2.1.4.3). Finally, §2.1.4.4 provides additional insights on additional (OOD) benchmarks and a comparison to using prompt ensembles and latent space noise. The suppl. material contains further experiments.

Implementation details. We utilize CLIP [155] as the underlying VLM for `WaffleCLIP`. As there is no direct cost associated with generating random character or word sequences, their number is only bounded by inference speed requirements (which is minimal as all respective language embeddings can be computed *a priori* [128]). However, we find diminishing returns for very high numbers (see also §2.1.4.4), and use 30 random descriptors per class (or 15 random character and word descriptor pairs) if not mentioned otherwise, with similar performance for both half or double the descriptor count (c.f. §2.1.4.4). All experiments use PyTorch [146] and are conducted on a single NVIDIA 3090Ti GPU. Wherever necessary, fine-grained LLM-generated descriptors are either taken from or generated following the codebase provided by [128]. If not mentioned explicitly, all results involving `WaffleCLIP` are computed over seven random seeds.

Benchmarks. The datasets considered are (mostly from [128]) ImageNet [42] and ImageNetV2 [99], CUB200-2011 [207] (fine-grained bird classification), EuroSAT [72] (satellite image recognition), Places365 [241], Food101 [17], Oxford IIIT Pets [145], DTD (Textures, [36]), Flowers102 [139], FGVCAircraft [122], and Stanford Cars [100].

High-level concepts. Following §2.1.3.3, the GPT-3 generated high-level concept for CUB200-2011 is `"Bird"`, `"Land Use"` for EuroSAT, `"Place"` for Places365, `"Food"` for Food101 and `"Breed"` for Oxford Pets. For additional benchmarks, extracted concepts are noted in section §2.1.4.4. For ImageNet (V2) and DTD, the concepts are too generic and thus filtered out (`"Object"`, `"Noun"`, or `"Adjective"`), with high-level guidance omitted.

2.1.4.1 `WaffleCLIP` vs LLM-generated descriptors

We start by analyzing the impact of randomization beyond fixed, randomized sets of fine-grained LLM-generated descriptors as done in Tab. 2.1, by instead using randomized character or word descriptors through our proposed `WaffleCLIP`. For that, we investigate visual classification accuracies across the eight diverse benchmarks studied in [128] in Tab. 2.2, where we compare `WaffleCLIP`, which does not use any external LLMs, with DCLIP. We find that averaging over randomized descriptors yields performances comparable to or better than those obtained with LLM-generated fine-grained descriptors over a majority of studied datasets, with average performance being comparable: 58.56% using DCLIP versus 58.57% for `WaffleCLIP` with a ViT-B/32 backbone and (see supp. material) 69.14% $\rightarrow$ 68.95% for ViT-L/14, and 54.77% $\rightarrow$ 54.20% for ResNet50. Beyond the inherently zero-shot nature of `WaffleCLIP` and ease of use, these results highlight that improved visual classification with pretrained VLMs does not require external LLMs, and further cements prompt averaging as a potential key driver behind DCLIP.

ViT-B/32	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	Flowers102	FGVCAircraft	Stanford Cars	Avg
CLIP [155]	54.71	62.01	51.28	40.78	39.12	82.59	85.06	43.18	62.89	24.99	58.54	55.01
DCLIP [128] (*mean*)	**55.82**	**63.12**	**52.47**	**43.29**	**40.47**	**82.79**	86.54	**43.99**	**64.01**	**26.94**	**57.08**	**56.05**
DCLIP [128] (*max*)	54.41	61.67	52.40	37.11	37.21	82.37	**88.03**	43.35	63.62	25.77	56.21	54.74

Table 2.3: **Importance of semantics in DCLIP.** Comparing similarity score averaging (*mean*) and maximum selection (*max*) reveals that taking the most similar entry even underperforms CLIP. This points to limited impact of LLM-generated semantics on improved visual classification.

ViT-B/32	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	Avg
DCLIP [128]	**55.82**	**63.12**	52.47	**43.29**	40.47	82.79	86.54	**43.99**	**58.56**
DCLIP (interchanged)	52.51 ±0.42	59.62 ±0.13	52.52 ±0.41	33.63 ±4.16	35.52 ±0.32	81.71 ±0.35	86.28 ±0.50	38.42 ±1.14	55.03 ±1.56
DCLIP (scrambled)	55.12 ±0.12	62.57 ±0.12	52.18 ±0.28	40.48 ±2.52	39.91 ±0.08	82.46 ±0.13	86.10 ±0.40	41.58 ±0.31	57.55 ±0.92
DCLIP (random, 1x)	54.11 ±0.28	61.37 ±0.18	52.42 ±0.19	36.83 ±4.27	38.80 ±0.26	82.86 ±0.23	85.99 ±0.62	42.20 ±0.85	56.82 ±1.57
DCLIP (random, 5x)	55.43 ±0.12	62.81 ±0.05	52.66 ±0.17	38.57 ±1.52	40.54 ±0.05	84.03 ±0.11	86.75 ±0.21	43.41 ±0.74	58.02 ±0.61

Table 2.4: **Progression from systematic to fully randomized descriptor scrambling.** To model systematic semantic shifts, we randomly swap descriptor lists between classes (*interchanged*), before progressing to shuffling descriptor words within the classes (*scrambled*) and randomly sampling LLM-generated descriptors for each class (*random*) from the complete set of descriptors with counts as in (or five times that of) DCLIP (*1x*, *5x*). As can be seen, a systematic shift results in a notable performance drop, while more independently randomized descriptors can recover the DCLIP performance, aligning with the observation that fully randomized prompt averaging is the main performance driver for `WaffleCLIP`.

2.1.4.2 Are descriptors from LLMs obsolete?

Our results above question the benefits of LLM-generated fine-grained semantics, as averaging over fully randomized character and word sequences achieves comparable performance. But does that mean that there is no benefit in leveraging descriptors produced by LLMs?

Impact of Averaging. To better understand this, we extend our motivational experiments from Tab. 2.1. First, we look at what happens when not performing averaging over all class descriptor distances as in DCLIP, but instead choosing the maximum. If additional fine-grained semantics were indeed beneficial, selecting the most suitable one should similarly raise the performance. However, as Tab. 2.3 reveals, performance actually drops, showing that the VLM cannot leverage the additional semantics to improve visual classification performance[2]. Instead, this again points to descriptor ensembling *as the main driver in performance*.

We further support this by studying additional descriptor randomization variants beyond those in §2.1.3.2. In particular, instead of swapping specific descriptors, we interchange full class-specific descriptor lists (*interchanged*). As descriptions often contain class-specific keywords, this models a systematic semantic shift away from the actual class. Additionally, we evaluate shuffling words within a descriptor list (*shuffled*), and descriptor lists subsampled from all available ones (*random*). This gives a progression from systematic to more independent descriptor randomization. Our results in Tab. 2.4

[2]This is potentially influenced by bag-of-words behavior of CLIP-like VLMs [229]. We leave the detailed analysis of this to future research.

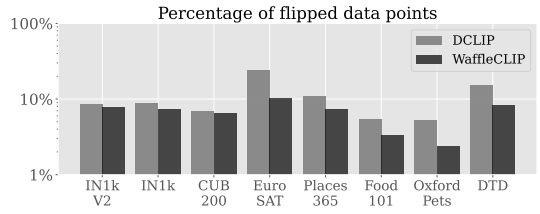


Figure 2.3: Label flipping experiment from CLIP to DCLIP or WaffleCLIP. Each bar indicates the percentage of data points getting either positively or negatively flipped (i.e. labelled correctly or incorrectly adjusted) when switching from CLIP to either DCLIP or WaffleCLIP. The consistently higher flip percentage indicates structural differences between natural language descriptors and randomized ones.

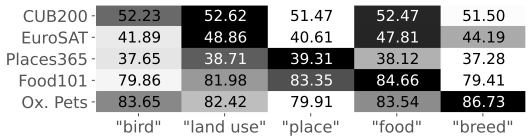


Figure 2.4: Study of semantic impact of GPT-3 generated high-level concepts. We find that interchanging the concepts generally reduces performance, indicating that high-level concepts provide complementary semantic context.

reveal that directly interchanging full *class-dependent* descriptor lists (*interchanged*) drops performance significantly (e.g. from 58.56% to 55.03% for ViT-B/32). In cases where no such shift is happening, we find performances to match DCLIP (e.g. 86.54% → 86.28% on Oxford Pets). Similarly, when moving from a systematic shift closer to fully randomized descriptors, performance approaches DCLIP (*scrambled* with 58.56% → 57.55% to *random* with 58.56% → 58.02%, see supp. material for more results). While this offers further evidence for WaffleCLIP and the fact that class-dependent ensembling drives gains, it does not yet allow us to directly compare the impact on the prediction behavior of LLM-generated descriptors and randomized ones.

Structural differences. We consider the percentages of samples that get positively or negatively flipped - i.e. classified correctly while previously being classified incorrectly (and vice versa) - when moving from CLIP to either DCLIP or WaffleCLIP in Fig. 2.3. We find that using LLM-generated fine-grained descriptors flips significantly more predictions than randomized words and characters, even when WaffleCLIP outperforms DCLIP. For example, DCLIP achieves 43.29% compared to WaffleCLIP with 44.31% on EuroSAT or 82.79% to 83.25% on Food101 in Tab. 2.2, but DCLIP flips a significantly larger portion of samples than WaffleCLIP. This reveals that full sentence, LLM-generated descriptors have a *structurally different* impact on the classification process, which we find to be *complementary* to randomization (see Tab. 2.2, + *GPT descr.*), where the use of both leads to additional improvements over WaffleCLIP (e.g. 58.57% → 60.21% for ViT-B/32).

This means that even if additional semantics are not the leading factor, LLMs for structured descriptor generation can still facilitate more robust class embeddings. Even *with* access to an external model for producing class descriptors, WaffleCLIP can provide additional benefits.

ViT-B/32	Flowers102	FGVCAircraft	Stanford Cars	Avg
CLIP [155]	62.89	24.99	58.54	48.81
DCLIP [128]	64.01	26.94	57.08	49.34
WaffleCLIP	66.27 ±0.26	25.66 ±0.19	58.91 ±0.17	50.28 ±0.21
+ Concepts	**67.19** ±0.19	28.44 ±0.22	**59.70** ±0.12	**51.78** ±0.18
+ GPT dsc. + Conc.	66.71 ±0.39	**28.96** ±0.37	59.33 ±0.14	51.67 ±0.32

Table 2.5: We find similar performance improvements with WaffleCLIP and high-level concept guidance for three additional benchmarks, which in parts do not benefit from LLM-generated descriptors (e.g. *Stanford Cars*).

Benchmarks	ImageNet-R [74]	ImageNet-S [208]	ImageNet-A [75]
CLIP [155]	65.97	40.73	29.63
DCLIP [128]	65.12	41.09	29.19
WaffleCLIP	**67.31**	**42.00**	**31.52**

Table 2.6: Performance gains of WaffleCLIP on distribution-shifted datasets further highlight general applicability through simple averaging over randomized descriptors, even if natural language ones fail.

2.1.4.3 Semantic guidance with high-level concepts

While we verified the relevance of additional semantic context through fine-grained descriptors, methods using additional fine-grained class information [128, 141, 152] suffer from the inherent ambiguities of some class names. As proposed in §2.1.3.3, our aim is to understand if high-level semantic context can be used to resolve such ambiguities by providing coarse semantic guidance for the class-retrieval process. Our results with extracted high-level concepts in Tab. 2.2, i.e. (+ *Concepts*), demonstrate consistent and significant improvements across most benchmarks and backbones when used with CLIP, with WaffleCLIP, and even alongside WaffleCLIP and DCLIP. These improvements are especially evident on benchmarks with ambiguous (e.g. Food101) or generic labeling (e.g. EuroSAT, with labels such as *Industrial* or *Residential*): For ViT-B/32, classification accuracy increases from 40.78% to 48.86% when applied to CLIP. Overall, the average classification accuracy also increases consistently (e.g. from 57.34% to 58.96% for ViT-B/32). This even beats DCLIP, despite only being applicable on five out of eight benchmarks (58.96% versus 58.56%). When applied to WaffleCLIP, improvements across most benchmark and backbone settings are also significant, although we find diminishing returns on the largest backbone, ViT-L/14, with average performance increasing only from 68.95% to 69.12% (see suppl. material). This might be due to its capabilities of retaining the most common concepts associated with specific classes, resulting in a robust class retrieval setup when averaging over multiple randomized descriptor variants.

We verify the benefits of high-level semantics further by looking at performance changes when concepts are interchanged (Fig. 2.4). For most benchmarks, the largest improvements are obtained with GPT-generated concepts. Some off-diagonal terms with higher scores, e.g. CUB200 where `"bird"` performs similar to/worse than `"land use"`/`"food"`, do appear out of distribution and warrant future research to improve our understanding of how semantics concepts are truly encoded in large VLMs.

However, seeing maximum performances primarily on the diagonal heuristically supports that additional semantics introduced as high-level concepts and commonalities, can offer reliable guidance. Indeed, considering a selection of ambiguous samples such as `"Boxer"` or `"Sphynx"` in the Oxford Pets dataset, `"Mussels"`, `"Oysters"` or `"Grilled Salmon"` in the Food101 dataset, or highly generic labels such as `"Industrial"`

ViT-B/32	IN1k-V2	IN1k	CUB	Euro	Places	Food	Pets	DTD	Flowers	FGVC	Cars	Avg
CLIP [155]	54.71	62.01	51.28	40.78	39.12	82.59	85.06	43.18	62.89	24.99	58.54	55.01
DCLIP [128]	55.82	63.12	52.47	43.29	40.47	82.79	86.54	**43.99**	64.01	26.94	57.08	56.05
P. Ensemble	55.49 ±0.21	62.79 ±0.29	51.46 ±0.43	45.76 ±0.49	40.58 ±0.06	82.67 ±0.37	83.26 ±0.72	42.53 ±0.54	63.30 ±0.33	25.14 ±0.45	58.38 ±0.29	55.58 ±0.42
+ Concepts	↓	↓	**52.08** ±0.17	**49.80** ±0.66	**40.61** ±0.14	**84.45** ±0.15	**87.42** ±0.20	↓	**65.38** ±0.27	**26.64** ±0.50	**59.12** ±0.14	**56.94** ±0.34
WaffleCLIP	**55.92** ±0.08	**63.31** ±0.09	52.38 ±0.12	44.31 ±1.07	40.56 ±0.07	83.25 ±0.21	85.70 ±0.25	43.16 ±0.25	66.27 ±0.26	25.66 ±0.19	58.91 ±0.17	56.31 ±0.37
+ Concepts	↓	↓	**52.83** ±0.19	**48.51** ±0.70	**40.97** ±0.08	**85.21** ±0.06	**87.52** ±0.10	↓	**67.19** ±0.19	**28.44** ±0.22	**59.70** ±0.12	**57.52** ±0.26

Table 2.7: **Prompt ensembling versus WaffleCLIP (+concepts)**. Across all visual classification benchmarks, we find matching or improved performance of WaffleCLIP compared to prompt ensembling (improving on eight out of eleven benchmarks), with the increase in average classification performance of WaffleCLIP compared to prompt ensembling higher than the increase of a prompt-ensembled version compared to standard CLIP, **without** requiring a handcrafted list of prompts.

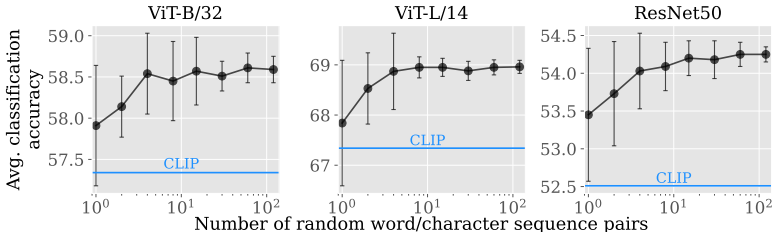


Figure 2.5: Ablation study on the number of randomized word and character descriptors used in WaffleCLIP. We find consistent competitive performance with just four randomized descriptor pairs (c.f. DCLIP Tab. 2.2). CLIP (blue line) is outperformed with just a single descriptor pair.

or "Residential" in the EuroSAT satellite image dataset, we find a consistent increase in average similarity to all associated test images by up to 13%. This confirms that concept guidance can re-align and refine class embeddings based on the relevant context.

2.1.4.4 Ablation studies

Evaluation on additional (OOD) benchmarks. We observe further evidence for the generality of WaffleCLIP and concept guidance by studying three additional benchmarks beyond those in Tab. 2.2 and [128]: Flowers102 [139] (extracted concept: "flower"), FGVCAircraft [122] ("aircraft"), and StanfordCars [100] ("car"). Our results in Tab. 2.5 (and in the suppl. material for other backbones) again show consistent gains when going from CLIP to WaffleCLIP or WaffleCLIP + *Concepts*. Interestingly, DCLIP is detrimental on very fine-grained benchmarks like Stanford Cars, losing 1.46% against CLIP. We speculate that this is due to semantically similar descriptors for multiple classes that are coarser than the actual class label (e.g. "BMW Active Hybrid" and "BMW 1 Series" being assigned similar generic BMW descriptors). Consequently, embeddings of related classes are systematically moved too close, harming performance. Meanwhile, WaffleCLIP (+ *Concepts*) can still offer performance boosts (58.54% → 58.91% → 59.70%). Furthermore, we study WaffleCLIP on OOD benchmarks: Adversarial natural images (ImageNet-A, [75]), sketches (ImageNet-S, [208]) and renditions (ImageNet-R, [74]). Results in Tab. 2.6

show that while DCLIP does not improve consistently, `WaffleCLIP` operates well even for out-of-distribution data (e.g. 29.63% → 31.52% on ImageNet-A).

Impact of randomization types. We analyze how performance changes when either using only random character sequences or only random word sequences instead of a combination of both as in `WaffleCLIP`. Across benchmarks and architectures (see Tab. 2.8), we observe dichotomies in performance between either random word or random character sequences, often performing either best or worst on a specific benchmark and backbone, while the joint usage of random words and character sequences strikes a consistent and best transferable average improvement across benchmarks and backbone architectures. Therefore, we chose the joint usage of both random words and characters as our default setup.

Avg.	ViT-B/32	ViT-L/14	RN50
Joint	58.57 ±0.41	68.95 ±0.18	54.20 ±0.23
Random Words	58.18 ±0.44	68.73 ±0.58	55.24 ±0.41
Random Characters	58.59 ±0.27	68.02 ±0.14	53.79 ±0.16

Table 2.8: **Randomized descriptor modes.** Considering the joint usage of randomized word and character sequences compared to only randomized word or character sequences, joint usage provides the most consistent performance improvements across benchmarks and backbones.

Comparison to latent noise. To highlight that input-level class-conditioned randomization is crucial, we compare to hyperspherical latent randomization. We use a von-Mises-Fisher distribution [40, 165, 209, 246]) to model hyperspherical unimodal distributions of class embedding vectors $\hat{\phi}^c$:

$$p(\hat{\phi}^c|\phi_l^c, \kappa) = C_d(\kappa)\exp(\kappa\phi_l^{cT}\hat{\phi}^c), \tag{2.3}$$

centered around a class centroid vectors ϕ_l^c with constant normalization $C_d(\kappa)$ only dependent on the dimensionality of ϕ and concentration κ. To sample from a vMF distribution around each class embedding, we leverage the sampler utilized in [40, 95] with the same budget of 30 noise embeddings. Average performance as a function of the (inverse) concentration κ is visualized in Fig. 2.6. For high concentrations (i.e. random embedding samples placed close to the mean direction), one can replicate the CLIP performance. For higher variances, performance continuously drops, with a hard inflection at around $\kappa \approx 10^4$. This shows that class-conditioned randomized descriptors as used in `WaffleCLIP` are crucial for providing a more robust estimate

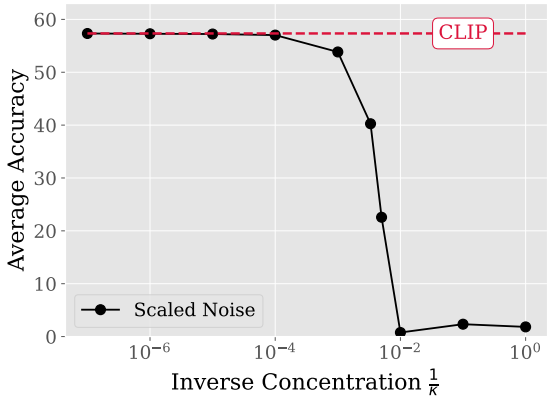


Figure 2.6: We compare latent space noise (vMF distribution around CLIP language embeddings) against standard CLIP. Reducing latent noise (i.e. increasing concentration κ) converges to initial performance, highlighting no notable benefit of deploying noise in the latent space.

of semantic concepts, and cannot be simulated through simple embedding space noise.

Comparison to prompt ensembles. We also compare WaffleCLIP to prompt ensembling (c.f. e.g. [152]) with the same budget of 30 randomly selected prompt options from a list of eighty handcrafted ones (taken from [152], such as `"A tattoo of a {class}."`, `"A {class} in a video game."`, ...). Unlike WaffleCLIP, prompt ensembling still requires human input and design. Results on all eleven benchmarks are listed in Tab. 2.7, which favor WaffleCLIP, outperforming prompt ensembling in eight out of eleven benchmarks and comparable performance on the remaining ones. In particular, improvements over prompt ensembling are higher than the improvement of prompt ensembling over vanilla CLIP (56.31% → 55.58% → 55.01%). This further supports the benefit of extracting more robust semantic representations, for which randomized descriptors provide a cheap and suitable tool. In addition to that, we highlight the complementarity of high-level concept guidance in combination with prompt ensembling in Tab. 2.7 (wherever the classname is included, we simply use `"a {concept}: a {classname}"` instead), raising the average classification accuracy from 55.58% to 56.94%.

Dependence on descriptor counts. We study the impact of the randomized word and character sequence pair count for WaffleCLIP in Fig. 2.5. A value of one indicates a single pair comprising a random words and characters descriptor, respectively. We achieve competitive performance already with 4 to 15 descriptor pairs (c.f. DCLIP in Tab. 2.2), while consistently outperforming CLIP (blue line) even with a single randomized descriptor pair. As class embeddings can be computed *a priori*, the impact on overall inference time is low, making WaffleCLIP and its extensions very attractive for enhancing image classification performance of VLMs.

2.1.5 Conclusion

In this work, we systematically examined the benefits of using LLM-generated class descriptors for improved training-free image classification with vision-language models (VLMs). In-depth studies reveal how similar performance gains can be achieved by replacing LLM-generated descriptors with randomized ones, giving rise to WaffleCLIP. Without access to external LLMs, across eleven visual classification benchmarks, we get comparable or better results than those obtained when using fine-grained GPT-3 generated descriptors. This makes WaffleCLIP very attractive for practical use in true zero-shot scenarios, and it serves as a crucial sanity check for future methods using external queries. We also show that VLMs struggle to leverage the actual semantics introduced through fine-grained semantic descriptors, and instead show that if given access to external LLMs, semantics are better exploited through coarse, high-level concepts. Using specific queries, we show how these can be automatically extracted, while jointly helping to address issues of class ambiguity.

2.2 Feasibility with Language Models for Open-World Compositional Zero-Shot Learning

Humans can easily tell if an attribute (also called state) is realistic, i.e., feasible, for an object, e.g. fire can be *hot*, but it cannot be *wet*. In Open-World Compositional Zero-Shot Learning, when all possible state-object combinations are considered as unseen classes, zero-shot predictors tend to perform poorly. Our work focuses on using external auxiliary knowledge to determine the feasibility of state-object combinations. Our Feasibility with Language Model (FLM) is a simple and effective approach that leverages Large Language Models (LLMs) to better comprehend the semantic relationships between states and objects. FLM involves querying an LLM about the feasibility of a given pair and retrieving the output logit for the positive answer. To mitigate potential misguidance of the LLM given that many of the state-object compositions are rare or completely infeasible, we observe that significant work needs to go into exploiting the in-context learning ability of LLMs. We present an extensive study on many prompt variants and involving six LLMs, including two LLMs with open access to the logit values, identifying Vicuna and ChatGPT as best performing, and we demonstrate that our FLM consistently improves OW-CZSL performance across all three benchmarks.

2.2.1 Introduction

Humans have the ability to discern the feasibility of state-object pairs, effortlessly distinguishing between realistic and implausible combinations. For instance, while it is convincing for a *fire* to be *hot*, the notion of a *wet fire* is nonsensical. Open-world compositional zero-shot learning (OW-CZSL) [123] seeks to emulate human-like understanding for compositional concepts. The task is to classify images to the correct state-object pair in the absence of explicit knowledge regarding the feasibility of the pairs in the candidate classes (referred to as open-world compositional zero-shot learning, OW-CZSL) when the model is trained with a small subset of feasible pairs. Models often struggle to achieve satisfactory performance in the open-world setting since all combinations of state-object pairs are considered prediction candidates, which includes both unseen feasible pairs and infeasible pairs, making the number of candidates much larger than the number of classes considered during training.

To address this challenge, prior works [86, 123] proposed to remove possibly infeasible pairs from the label space using word vectors such as GloVe [148] or using external resources such as ConceptNet [191]. While these approaches represent a step forward, open-world compositional zero-shot learning remains extremely challenging as these approaches are limited in their capability to capture the semantic relationships underlying many rare concept compositions. Therefore, our goal is to propose a more effective approach for determining the feasibility of state-object pairs even if they are rare.

Large language models (LLMs) recently demonstrated strong language comprehension

capabilities across various NLP tasks [239]. In this work, we propose Feasibility with Language Model (FLM) to predict the feasibility score of any state-object pair, with the purpose of better aligning with the human-annotated ground-truth feasibility than previous approaches. Concretely, we ask an LLM to give a binary response, i.e., "Yes" or "No", indicating the feasibility of the given state-object pair. The output logit for the positive answer would then be considered the feasibility score for the corresponding pair.

Inevitably, one challenge in using LLMs for feasibility prediction is that, provided without context a query could lead to many false negatives. Consider the “dark fire” class of the MIT-States [82] dataset, that is considered feasible. Asking a LLM, whether “dark fire” exists, yields the answer “No”, presumably because the state “dark” is not typically associated with bright objects. However, “dark fire” is a reasonable class in MIT-States, as humans assigned this label to highlight the dark surroundings and dim visual theme for these images of fire. To teach the LLM about the relevant context for image classification, we can inform the LLM of semantically similar and feasible compositions from the training set, such as “dark lightning”. As a result, the LLM can correctly infer *in-context* that the state "dark" can also be associated with “fire”.

To summarize, our contributions are: 1) in Feasibility with Language Model (FLM) we propose to leverage LLMs to predict the feasibility of state-object pairs in open-world CZSL where we provide guided prompts that include examples of true feasible pairs, enhancing the LLMs’ understanding of the CZSL task via in-context learning, 2) while the compositional label’s feasibility judgement via FLM better align with human-annotated ground truth, FLM can also be integrated into any existing VLM, 3) FLM consistently improves CZSL performance over previous state-of-the-art methods on all three challenging benchmark datasets.

2.2.2 Related work

CZSL. CZSL aims to classify instances of state-object compositions not seen during training. Early approaches tackle CZSL by employing two separate classifiers for state and object primitives, followed by an additional module composing them to generate a classifier for the composition pairs [130, 153]. Another line of research treats the state as a transformation operator that modifies attributes of the object, and trains models to satisfy properties like symmetry or commutativity [108, 134]. Additionally, GCNs have been leveraged to capture the relevance between states, objects, and state-object pairs, enhancing the models’ generalization ability to unseen classes [132, 167].

VLMs for CZSL. VLMs like CLIP [155] have been employed to address CZSL. CSP [135] introduces a parameter-efficient learning technique, which fine-tunes the prompt, similar to CoOp [243], but it updates the tokens representing states and objects instead of the prefixed context tokens. CSP utilizes vision-language understanding while adapting specifically to the downstream CZSL task, outperforming previous CNN-based task-specific methods.

OW-CZSL. Initially, CZSL models were evaluated exclusively on unseen classes. Then [24] introduced a generalized setting that considers both seen and unseen classes as potential labels, which was first used in CZSL by [153]. The open-world setting [123], extends the output space to include all possible combinations of states and objects, which is a more challenging task due to the substantial increase in the number of potential candidates. Consequently, identifying and disregarding infeasible state-object pairs becomes crucial. Prior approaches have employed word vectors, i.e., CompCos [123], or external knowledge, i.e., KGSP [86], to determine the feasibility of pairs. In contrast, we propose to leverage state-of-the-art large language models to capture the feasibility of pairs more effectively.
LLMs as guidance. LLMs have solved many NLP tasks, e.g. GPT-3 [19], ChatGPT [144], PaLM 2 [5], Claude 2 [6] LLaMa 2 [199, 200] and Vicuna [32] demonstrate exceptional language understanding capabilities and have been widely applied across diverse NLP downstream tasks. It was shown by [19] that LLMs can perform in-context learning without requiring fine-tuning, i.e., LLMs can significantly improve on a given task through task-specific exemplars demonstrating how a task is performed. On top of few-shot examples, providing explanations in-context further boost task performance on challenging tasks [101]. Additional context can also be directly provided by the LLM. With chain-of-thought prompting [212], the in-context demonstrations provide more elaborate answers, resulting in the same behaviour of the LLM and ultimately more accurate responses. Similarly, [7] find that aggregating multiple prompting structures helps alleviate the sensitivity of LLMs in-context prompt design.

2.2.3 In-context Feasibility Prediction Framework

In this section, we first describe the general setting of compositional zero-shot learning in §2.2.3.1. Next, we explain the novel utilization of large language models for predicting feasibility scores in §2.2.3.2.

2.2.3.1 Open-World Compositional Zero-Shot Learning (OW-CZSL)

CZSL aims to classify an image, where each class is a state-object combination. Given a set of states $\mathcal{S}$ and objects $\mathcal{O}$, the label space of a training set $\mathcal{Y}_{seen}$ corresponds to the subset of all possible pairs of state and object, $\mathcal{Y}_{seen} \subset \mathcal{Y}_{all}$, where $\mathcal{Y}_{all} = \{(s, o)|s \in \mathcal{S} \text{ and } o \in \mathcal{O}\}$. The model is trained on the training set with candidate labels as seen classes $\mathcal{Y}_{seen}$, and the goal of CZSL is to classify an image from the test label space $\mathcal{Y}_{test}$ that contains both seen and unseen classes, $\mathcal{Y}_{test} = \mathcal{Y}_{seen} \cup \mathcal{Y}_{unseen}$, where $\mathcal{Y}_{unseen}$ denotes the unseen classes during the training, $\mathcal{Y}_{seen} \cap \mathcal{Y}_{unseen} = \emptyset$ and $\mathcal{Y}_{unseen} \subset \mathcal{Y}_{all}$.

In the closed-world setting, it is assumed that we have prior knowledge of feasible sets, i.e., the label space at test time is restricted to $\mathcal{Y}_{test} := \mathcal{Y}_{seen} \cup \mathcal{Y}_{unseen}$ and known to the model. The open-world setting assumes no prior information about the set of unseen compositions at test time, i.e., $\mathcal{Y}_{test} := \mathcal{Y}_{all}$. The substantial increase in the label space in the open-world setting leads to a significant performance gap compared to

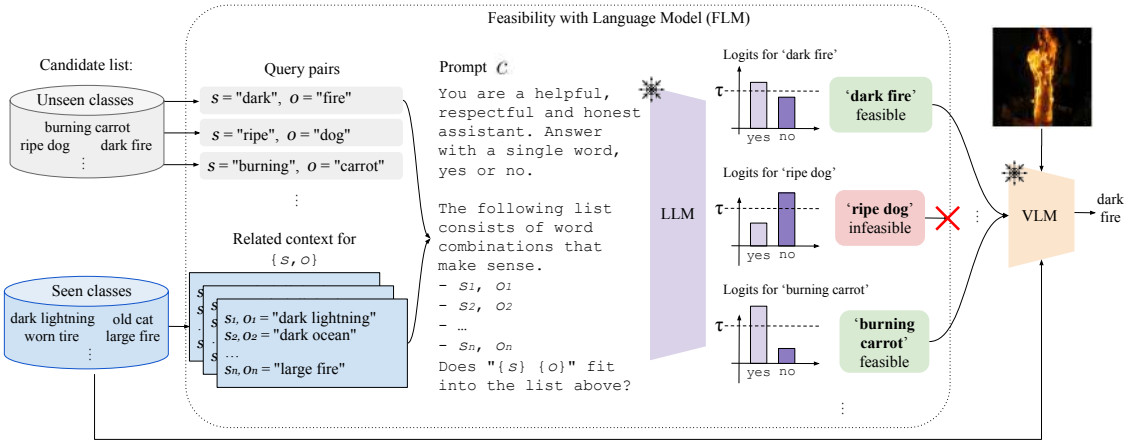


Figure 2.7: The pipeline of our Feasibility with Language Model (FLM) method. We constrict a prompt containing a list of related seen classes from the training set and a query to classify an unssen state-object pair as feasible. By comparing the LLM logit for the token "Yes" with a threshold τ we determine whether a pair is feasible, in which case it is used for OW-CZSL classification.

the closed-world setting. To mitigate this performance gap, previous works [86, 123] have developed a function $g(\cdot)$ that assigns a feasibility score to each class, indicating the likelihood of its feasibility. By setting a threshold τ, classes with scores below the threshold are deemed infeasible and consequently removed from the test label space, $\mathcal{Y}_{test} := \{y | y \in \mathcal{Y}_{all} \text{ and } g(y) \geq \tau\}$.

It is important to note that an accurate feasibility function plays a critical role in open-world CZSL. If the function assigns low feasibility scores to truly feasible classes, the model will fail to predict the correct class as it is absent from the test label space. Conversely, if the function assigns high feasibility scores to numerous infeasible classes, it becomes more likely that the model will make incorrect predictions.

2.2.3.2 Feasibility with Language Model (FLM)

LLMs are autoregressive models that generate words by sampling from a predicted probability distribution, i.e., they model $p_{\text{LLM}}(t_k | t_1, \ldots, t_{k-1})$ where t is a token from the vocabulary of the language model. In other words, the output probability indicates how certain the LLM is that a given token should appear next. Essentially, our FLM uses the output of an LLM as a measure of feasibility.

Canonical prompt. To obtain feasibility scores using LLMs, we construct a prompt c that consists of a system message, *sysmsg*, and a human message, *hmsg*. The *sysmsg* provides the LLM with general guidance while the *hmsg* asks the LLM to assess the feasibility of a given class.

$$c = \{sysmsg: \text{"You are a helpful, respectful and honest assistant. Answer with a single word, yes or no."},$$
$$hmsg: \text{"Does a/an \{s\} \{o\} exist in the real world?"}\}$$

where we refer to the first sentence of the *sysmsg* as the *persona* component [172], the second as the *instruction* component, and the sentence of the *hmsg* as the *query* component. The placeholders $\{s\}$ and $\{o\}$ represent the state and object of the class, respectively.

The output probability distribution generated by the LLM reflects its level of certainty regarding the occurrence of specific words. In our case, the probability or the logit of the word "Yes" in the output distribution indicates the LLM's confidence in the feasibility of the given pair (s, o). We interpret this output as a feasibility score. More formally, our feasibility score function is

$$g(s, o) = \log p_{\text{LLM}}\left(t = \texttt{"Yes"} | f(s, o; c)\right) \tag{2.4}$$

where $\log p_{\text{LLM}}$ indicates the unnormalized output logits and $f(s, o; c)$ denotes a function that composes the prompt c with the target state-object pair (s, o). To obtain a real-valued score, this method requires local access to the LLM. When an LLM is accessed through an API, such as ChatGPT, it might not expose the output probabilities or logits of the model, i.e. $\log p_{\text{LLM}}\left(t = \texttt{"Yes"} | f(s, o; c)\right)$. In this case we can only retrieve a binary score of "Yes" or "No".

In-context learning. There is potential for incorrect responses when simply querying the LLM about the feasibility of a given pair, e.g. "dark fire" as mentioned in §2.2.1. Motivated by this, we leverage the in-context learning capabilities of LLMs. In-context learning enables LLMs to adapt to new tasks with minimal examples. In addition to the *query* component, we introduce a *guidance* component to the human message. The guidance includes a few examples of true feasible pairs, allowing the LLMs to learn from these instances and better understand what constitutes feasibility within the dataset. For example, a human message with guidance would be:

"`The following list consists of word combinations that make sense.`
– $\{s_1\}$ $\{o_1\}$
– ...
– $\{s_n\}$ $\{o_n\}$
`Does "{`s`} {`o`}" fit into the list above?`"

where $\{(s_i, o_i)\}_{i=1}^{n}$ are few-shot examples of true feasible pairs. Several approaches can be employed to select examples for the guidance prompt. One straightforward method is to randomly sample pairs from the seen classes $\mathcal{Y}_{seen}$. Another approach is to leverage the information from the query pairs. Motivated by [123], we choose guidance pairs from the seen classes that either include the state s or the object o, i.e., $\mathcal{Y}_{pos} = \{(s_i, o_i) | (s_i, o_i) \in \mathcal{Y}_{seen},\ s_i = s \text{ or } o_i = o\}$. This strategy enables the LLMs to gain a deeper understanding of the dataset-specific task within the in-context learning framework, improving predictions of the feasibility of query pairs. The overall pipeline is drawn in Figure 2.7. Our feasibility score function, denoted as Feasibility with Language Model (FLM), is formulated as

$$g(s, o) = \log p_{\text{LLM}}\left(t = \texttt{"Yes"} | f(s, o, \mathcal{Y}_{pos}; c)\right) \tag{2.5}$$

where $f(s, o, \mathcal{Y}_{pos}; c)$ denotes a function that composes the prompt c with the target state-object pair (s, o) and the related seen pairs $\mathcal{Y}_{pos}$.

Versatility. Once we obtain the feasibility scores for all combinations of pairs, the threshold τ determines the subset of all pairs that are deemed feasible. The infeasible pairs are discarded, and only the feasibile pairs are used as candidate labels for the VLM's prediction. Our feasibility scores can be integrated with any existing VLM to improve performance in the open-world setting.

2.2.4 Experiments

We present our experimental findings on LLM-guided feasibility prediction in OW-CZSL. The experimental setup is detailed in §2.2.4.1 and Appendix, followed by a quantitative and qualitative comparison with baselines in §2.2.4.2. We evaluate the feasibility prediction in isolation in §2.2.4.3, compare a variety of LLMs in §2.2.4.4, and conduct an ablation study in §2.2.4.5, examining the different prompt components, such how the number of guidance examples influence the results.

2.2.4.1 Experimental setup

Benchmarks. We use three standard datasets for OW-CZSL, i.e., MIT-States [82], UT-Zappos [225, 226], and C-GQA [132]. Each dataset comprises a set of states and objects, where an object-state combination forms a class.

Evaluation metric. We follow the protocol of [153] for OW-CZSL. Since the VLM is trained only on seen classes, it is prone to being biased towards classifying an image as one of the seen classes at test time. Concretely, a calibration bias is subtracted from the model outputs of the seen classes, and then the class is predicted. The calibration bias is varied to get the best combination of seen class accuracy (denoted as S), unseen class accuracy (U), harmonic mean of accuracy on seen class and unseen class (H), and area under the curve of seen class and unseen class accuracy (AUC). By tackling the feasibility prediction of unseen classes, we focus on improving the more challenging metrics (U, H, AUC) while the seen class accuracy (S) remains unaffected.

Implementation details. For OW-CZSL, hyperparameters are traditionally explored, and the best model is chosen based on the highest unseen validation accuracy. We perform a grid search on sentence variations as well as choose the threshold τ that determines whether the query class is feasible by the best unseen validation accuracy (more details in the Appendix). For the LLM, we use the Vicuna-13B model [32] for the experiments unless otherwise indicated.

Feasibility baselines. We compare with GloVe embeddings [148] as used in Comp-Cos [123] and CSP [135], and the ConceptNet [191] as used in KGSP [86]. For GloVe, the cosine similarity between the concepts of the same primitives are calculated and merged to represent the feasibility score. ConceptNet [191] is a knowledge graph connecting

ViT-L/14		MIT-States				UT-Zappos				C-GQA			
VLM	Method	S	U	H	AUC	S	U	H	AUC	S	U	H	AUC
CLIP	GloVe	30.21	14.6	13.0	3.10	10.8	19.3	10.6	1.60	7.59	**3.92**	2.46	**0.20**
	ConceptNet	—"—	12.5	12.7	2.75	—"—	21.3	10.3	1.69	—"—	2.01	2.59	0.13
	FLM (ours)	—"—	**16.1**	**13.7**	**3.38**	—"—	**23.6**	**11.5**	**1.94**	—"—	2.62	**2.82**	0.16
CoOp	GloVe	38.2±0.9	16.7±0.3	16.2±0.4	4.78±0.2	61.2±1.8	36.7±2.4	34.2±3.7	18.1±2.8	26.9±1.6	5.09±0.5	6.16±0.5	1.03±0.1
	ConceptNet	—"—	14.5±0.3	15.6±0.3	4.36±0.1	—"—	42.1±2.3	36.6±3.2	20.6±2.5	—"—	4.14±0.5	5.88±0.7	0.92±0.2
	FLM (ours)	—"—	**18.7**±0.3	**17.4**±0.5	**5.40**±0.1	—"—	**49.6**±1.7	**40.6**±3.1	**24.4**±2.7	—"—	**5.16**±0.3	**6.91**±0.3	**1.13**±0.1
CSP	GloVe	45.1±0.9	14.9±0.3	16.4±0.4	5.12±0.2	62.8±0.9	45.8±1.8	38.9±0.8	22.6±1.0	30.2±0.5	**4.58**±0.5	6.12±0.5	1.09±0.1
	ConceptNet	—"—	13.4±0.8	15.5±0.5	4.74±0.3	—"—	54.0±1.7	43.3±0.9	26.9±1.1	—"—	1.31±0.1	2.25±0.3	0.34±0.0
	FLM (ours)	—"—	**16.6**±0.3	**17.4**±0.6	**5.76**±0.2	—"—	**56.7**±1.3	**43.9**±0.9	**30.0**±1.1	—"—	4.55±0.5	**6.55**±0.5	**1.13**±0.1

Table 2.9: CSZL results comparing Glove, ConceptNet and our FLM (Vicuna, logit) on MIT-States, UT-Zappos and C-GQA. We report seen (S) and unseen class accuracy (U), harmonic mean (H) and AUC using the CLIP, CoOp and CSP as base models. Ditto (—"—) denotes "same as above".

words to obtain the feasibility scores which are calculated by the cosine similarities of ConceptNet embeddings.

2.2.4.2 LLM-generated feasibility in OW-CZSL

We evaluate our FLM method using three VLMs: CLIP [155], CoOp [243], and CSP [135], which are CLIP-based models. We choose VLMs since they outperform the CNN-based task-specific CZSL methods, as reported by [135]. We use ViT-L/14 as the VLMs' backbone and run the CSP official code[3] with fixed hyperparameter settings for CoOp and CSP (details in Appendix), although optimizing these hyperparameters as done by [135] could further yield improvements. We run CoOp and CSP with 5 different seeds and report the mean and the standard deviation. CLIP is applied without any fine-tuning, and thus standard deviation is not reported.

Quantitative comparison. The experimental results are presented in Table 2.9 where we improve OW-CZSL performance across various scenarios and on all datasets. We first observe that on the MIT-States dataset, our FLM method achieves the highest harmonic mean of 17.4% and AUC of 5.76%, surpassing the GloVe feasibility function which shows 16.4% and 5.12%, and the ConceptNet feasibility function with 15.5% and 4.74% using the CSP model. FLM exhibits even more significant improvements on the UT-Zappos dataset. When compared to the GloVe feasibility, we observe a substantial 7.4% increase in AUC (from 22.6% to 30.0%) for the CSP model, a 6.3% increase (from 18.1% to 24.4%) with the CoOp model, and a 0.34% increase (from 1.60% to 1.94%) in the CLIP model. Similarly, FLM outperforms ConceptNet on UT-Zappos. On the C-GQA dataset, FLM performs the best everywhere except on U and AUC metrics in the CLIP model. However, CLIP is not the best model for OW-CZSL and generally falls behind CoOp and CSP. Overall, FLM achieves the best results on all metrics across all datasets. These results indicate

[3] https://github.com/BatsResearch/csp

Figure 2.8: Feasible examples from the unseen test set along with feasibility scores normalized such that the threshold τ is at 0. Positive scores (green) indicate a correct prediction as feasible, while negative scores (red) incorrectly infer infeasibility of a pair. Red box includes failure cases of FLM.

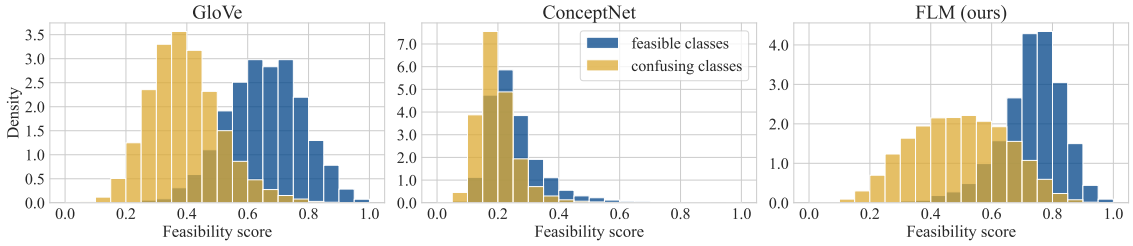


Figure 2.9: Distributions of feasibility scores of all state-object pairs. For best separation, feasible classes should be close to 1 and all remaining confusing classes close to 0.

that FLM can better differentiate between feasible state-object pairs and infeasible ones, as it facilitates all base OW-CZSL models to obtain a higher score, closing the gap to the closed-world setting.

Qualitative comparison. In Figure 2.8, we show qualitative results of feasible images from the unseen classes alongside the absolute difference of the feasibility score from the threshold. Positive values (green) indicate a correctly identified pair, while negative values (red) indicate an incorrect feasibility prediction. For each dataset, we show examples comparing FLM with GloVe and ConceptNet. For instance, FLM correctly identifies that "ruffled chair" is feasible for the MIT-States dataset, and that "teal armchair" is feasible in the context of the C-GQA dataset, both of which are considered infeasible by GloVe and ConceptNet. By providing seen pairs that are relevant to the query pair in the guidance prompt, e.g. "ruffled bed" for the query "ruffled chair" and "tan armchair" for the query "teal armchair", our FLM correctly identifies the given query pairs as feasible.

2.2.4.3 Ablation Study: Feasibility Prediction in Isolation from the OW-CZSL Task

We evaluate the feasiblity prediction in isolation from the OW-CZSL task and analyze the distributions of feasibility scores on the C-GQA dataset. In Figure 2.9, the unseen classes are referred to as "feasible classes" (blue) and the classes that are absent in both

	MIT-States				UT-Zappos				C-GQA			
	Feas. acc.	Infeas. acc.	Arith. mean	H. mean	Feas. acc.	Infeas. acc.	Arith. mean	H. mean	Feas. acc.	Infeas. acc.	Arith. mean	H. mean
GloVe	51.7	**93.2**	72.5	66.5	78.8	38.2	58.5	51.4	40.0	**98.5**	69.2	56.9
ConceptNet	52.0	92.5	72.3	66.6	**100.0**	13.2	56.6	23.3	26.3	91.7	59.0	40.9
FLM (ours)	**64.7**	86.1	**75.4**	**73.9**	93.9	**46.1**	**70.0**	**61.8**	**70.9**	89.2	**80.1**	**79.0**

Table 2.10: Accuracy of correctly identifying feasible and infeasible open-world pairs from $\mathcal{Y}_{all}$ using the same threshold τ as in Table 2.9. FLM uses Vicuna (logit) as LLM.

the seen and unseen sets are referred to as "confusing classes" (orange). Note that the scores obtained from each method have been normalized to fall within the range of 0 and 1.

FLM exhibits a better separation between the two distributions than GloVe and ConceptNet implying that our approach more effectively distinguishes between feasible and infeasible classes, providing an accurate assessment of the feasibility. It is important to note that many combinations contained in the "confusing classes" can still be realistically feasible, but simply not included in the dataset. Therefore, is is unlikely these distributions can be perfectly separated. By employing an appropriate threshold, our FLM method includes a greater number of feasible classes while significantly reducing the inclusion of infeasible classes among the candidate labels for open-world CZSL.

We evaluate overlap of the feasibility scores with the human annotations quantitatively in Table 2.10. For every state-object pair in $\mathcal{Y}_{all}$, we compute the feasibility prediction of GloVe, ConceptNet, and FLM and compare it to the human-annotated ground truth of feasible classes $\mathcal{Y}_{unseen}$. We report feasibility accuracy as the ratio of pairs in $\mathcal{Y}_{unseen}$ correctly identified as feasible and, analogously, infeasible accuracy for the ratio of all other classes predicted as infeasible. As we have seen above, the distributions of feasible and infeasible pairs are not perfectly separable and, thus, these two metrics form a trade-off that can be varied using the threshold value τ. The reported metrics are calculated using the same threshold values as in Table 2.9. We observe that our FLM performs the best on either feasible accuracy or infeasible accuracy, which suggests that the best trade-off between feasible and infeasible accuracy varies by dataset. Considering both metrics together through arithmetic and harmonic means, our method performs the best across all datasets, often by a significant margin, such as improving harmonic mean over GloVe by +10.4% on UT-Zappos and +22.1% on C-GQA. Consequently, the more accurate prediction of feasibility scores, which better align with the human-annotated ground truth, results in FLM performing better than GloVe and ConceptNet.

2.2.4.4 Ablation Study: Comparing Large Language Models

We use ChatGPT [19, 144], GPT-4 [143], PaLM-2 [5], Claude-2 [6], LLaMa-2-Chat-13B [200] and Vicuna-13B [32]. Vicuna-13B and LLaMa-2-Chat-13B are open-source language models fine-tuned to follow instructions that reach similar capabilities to ChatGPT in some

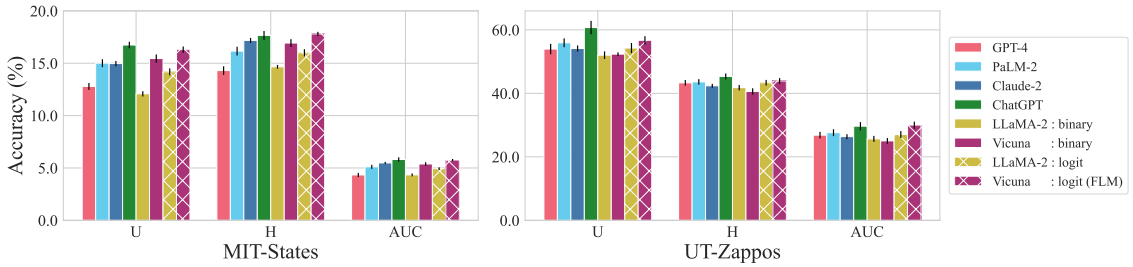


Figure 2.10: Comparison of FLM using Vicuna, LLaMA-2, and proprietary models (GPT-4, PaLM-2, Claude-2, and ChatGPT) as LLMs. Proprietary models can only provide a binary "Yes" or "No" response, whereas for Vicuna and LLaMA-2 we evaluate both the binary and logit outputs as feasibility scores.

settings. For brevity, we refer to these models as "Vicuna" and "LLaMa-2" in the following. One advantage of Vicuna and LLaMa-2 over proprietary models is the accessibility of internal values such as the probability or logit of the output words. Specifically, we utilize the logit value of the word "Yes" as our feasibility score. Moreover, we compare with ChatGPT, GPT-4, PaLM-2 and Claude-2 as proprietary LLMs where we query the API to obtain binary feasibility scores, i.e., a score of 1 when the model answers with "Yes" and 0 when it answers with "No". In this case, the threshold τ cannot be varied and is set to 0.5.

Proprietary LLMs such as ChatGPT [19, 144] often demonstrate superior performance compared to open-source models such as Vicuna [32] and LLaMa-2. However, there are distinctions between the two types of models in terms of accessibility. To ensure consistency and eliminate randomness, we set the temperature parameter to 0 during these experiments. Due to API constraints, we conduct these experiments only on the MIT-States and UT-Zappos datasets. The results for GPT-4, PaLM-2, Claude-2, ChatGPT, LLaMa-2, and Vicuna are depicted in Figure 2.10.

Across both datasets, we observe consistent trends. Firstly, both Vicuna and LLaMa-2 show lower performance with a binary answer compared to using logits on all three evaluation metrics. This suggests that accessing the logits provides valuable information for estimating feasibility. Among these two models, Vicuna outperforms LLaMa-2 clearly. Secondly, among the proprietary LLMs, ChatGPT performs best, surpassing PaLM-2, Claude-2 and even GPT-4. The differences are more pronounced on MIT-states than on UT-Zappos where most LLMs tend to perform similarly. Lastly, ChatGPT with a binary answer consistently outperforms Vicuna with a binary answer and oftentimes achieves even better results than Vicuna using logits. From these findings, we speculate that ChatGPT with logit access would likely surpass Vicuna with logit considerably, implying that more advanced LLMs with logit access would yield improved feasibility scores and, thus, could further push state-of-the-art in OW-CZSL.

		MIT-States			UT-Zappos			C-GQA		
Prompt		U	H	AUC	U	H	AUC	U	H	AUC
Canonical		13.5±0.4	15.4±0.3	4.70±0.2	47.1±1.4	39.2±0.6	23.1±0.6	2.16±0.5	3.40±0.6	0.52±0.1
Instruction: hmsg begin		15.5±0.4	16.6±0.3	5.32±0.2	58.2±1.2	44.1±0.6	28.3±0.7	3.06±0.4	4.42±0.3	0.69±0.0
Instruction: hmsg last		14.8±0.3	16.4±0.3	5.11±0.1	56.6±1.2	43.8±0.9	27.9±1.1	2.84±0.1	4.23±0.3	0.66±0.1
Format	QA: yes	10.3±0.7	12.0±0.5	3.39±0.3	53.1±1.6	42.4±1.3	26.7±1.2	2.23±0.3	3.40±0.6	0.49±0.1
	QA: score	10.8±0.3	12.3±0.3	3.53±0.1	54.2±1.6	43.4±0.9	26.9±1.0	2.36±0.1	3.69±0.1	0.54±0.0
In-context	$\mathcal{Y}_{pos}$, 5	13.8±0.3	15.3±0.3	4.74±0.1	48.2±8.0	39.9±4.0	24.0±4.2	3.16±0.3	4.67±0.6	0.76±0.1
	$\mathcal{Y}_{pos}$, 20	14.7±0.6	16.3±0.3	5.12±0.1	56.7±1.3	43.9±0.9	30.0±1.1	2.66±0.5	4.00±0.7	0.63±0.1
	$\mathcal{Y}_{pos}$, 50	16.6±0.3	17.4±0.6	5.76±0.1	—"—	—"—	—"—	3.25±0.5	4.62±0.7	0.76±0.1
	$\mathcal{Y}_{pos}$, 200	—"—	—"—	—"—	—"—	—"—	—"—	4.55±0.5	6.55±0.5	1.13±0.1
	random, 200	13.2±0.6	14.9±0.7	4.50±0.3	50.4±0.9	40.7±0.7	24.8±0.7	2.95±0.3	4.51±0.6	0.72±0.1
FLM (ours)		**16.6**±0.3	**17.4**±0.6	**5.76**±0.1	**56.7**±1.3	**43.9**±0.9	**30.0**±1.1	**4.55**±0.5	**6.55**±0.5	**1.13**±0.1

Table 2.11: Ablation study. Experiments are done with CSP as VLM model. We ablate the canonical query without in-context exemplars, placing instruction component in the human message, a different format of the guidance component, varying the number of in-context pairs, and a random ordering of in-context examples. Ditto (—"—) denotes the result is the same as the previous line.

2.2.4.5 Ablation Study: Analysis of LLM-Prompts

Comparison of instruction prompts. We investigate the impact of the prompt by comparing the performance of our in-context learning prompt with two ablations: 1) the canonical prompt described in §2.2.3.2 which does not use in-context examples, and 2) placing instruction component, e.g. "`Answer with a single word, yes or no.`", in the human message instead of the system message. The results on the three datasets are presented in the first three rows of Table 2.11.

Across all datasets, we observe that using the canonical prompt significantly drops the performance, e.g. 17.4%→15.4%, 43.9%→39.2%, 6.55%→3.40% in harmonic mean on MIT-States, UT-Zappos, and C-GQA, respectively. This highlights the importance of providing in-context guidance in our FLM for feasibility prediction. Moreover, placing an instruction in the human message, whether at the beginning or end, drops the performance on MIT-States and C-GQA while showing similar performance on UT-Zappos. This suggests that incorporating an instruction in the system message effectively guides the LLMs to the desired behavior.

Format for in-context learning. [7] have demonstrated the effectiveness of LLMs' in-context learning when using a question-and-answer format. To investigate this approach for FLM, we use a question-answer format for the guidance prompt instead of a list. Specifically, we employ the guidance prompt "`Does a/an {`s_i`} {`o_i`} exist in the real world? Yes.`", which is repeated for every related seen pair in $\mathcal{Y}_{pos}$, and the query prompt "`Does a/an {`s`} {`o`} exist in the real world?`" while keeping the rest of the process the same. The results are shown in the "Format QA: yes" row of Table 2.11.

We observe the performance drops across datasets, e.g. 17.4%→12.0 and 6.55%→3.69% in harmonic mean on MIT-States and C-GQA. The lower performance originates from this prompt format biasing the LLM to answer "Yes" because we only have access to feasible pairs. We observe a similar trend for "Format QA: score" where we use the same question-answer format, but instruct the LLM to respond with an integer score indicating the level of feasibility (see Appendix for details). Both of these results indicate that employing our proposed list format is crucial in obtaining accurate feasibility scores because we do not have access to infeasible examples.

Number of pairs for in-context guidance. To analyze the influence of the number of in-context examples in the guidance prompt, we conducted experiments varying the number of positive pairs. Recall that our FLM method selects related pairs in the guidance as $\mathcal{Y}_{pos} = \{(s_i, o_i) | (s_i, o_i) \in \mathcal{Y}_{seen},\ s_i = s \text{ or } o_i = o\}$. The performance results are presented in the "in-context" rows of Table 2.11.

Across all datasets and evaluation metrics, performance consistently improves as the number of pairs in the guidance increases. For instance, on MIT-States, the harmonic mean increases from 15.3% to 16.3%, and subsequently to 17.8%, as the number of pairs expands from 5, to 20, and to 50, respectively. Each dataset contains a different maximum number of related seen pairs. Thus, performance does not improve beyond 50 for MIT-States and beyond 20 for UT-Zappos. Moreover, using up to 200 randomly selected state-object pairs results in worse performance than just 5 related pairs from $\mathcal{Y}_{pos}$ on MIT-States and C-GQA. This suggests that it is important to provide relevant in-context pairs and that more few-shot examples allow the LLM to better comprehend the context-dependent task, leading to more accurate feasibility scores.

2.2.5 Conclusion

In this paper, we proposed a novel approach that leverages large language models (LLMs) to predict the feasibility of the state-object pair for the open-world compositional zero-shot learning (OW-CZSL). By leveraging the autoregressive nature of LLMs, we designed prompts to query the feasibility of class pairs to LLMs, and obtain the output of the word "Yes" which we consider as feasibility score. We used the in-context learning capabilities of LLMs by providing guidance prompts that included a few examples of true feasible pairs. Our experimental results validated the effectiveness of our LLM-guided feasibility approach. We compared our FLM method with previous approaches and achieved better performance in MIT-States, UT-Zappos, and C-GQA datasets. Through an analysis of the feasibility score prediction, we demonstrated that our Feasibility with Language Model effectively differentiated between feasible and infeasible classes when compared to human-annotated ground truth. Furthermore, ablation studies on the prompt setting revealed that the in-context learning framework with seen pairs as a guide was a key factor to having high-quality feasibility scores.

3

DataDream: Few-shot Guided Dataset Generation

While text-to-image diffusion models have been shown to achieve state-of-the-art results in image synthesis, they have yet to prove their effectiveness in downstream applications. Previous work has proposed to generate data for image classifier training given limited real data access. However, these methods struggle to generate in-distribution images or depict fine-grained features, thereby hindering the generalization of classification models trained on synthetic datasets. We propose DataDream, a framework for synthesizing classification datasets that more faithfully represents the real data distribution when guided by few-shot examples of the target classes. DataDream fine-tunes LoRA weights for the image generation model on the few real images before generating the training data using the adapted model. We then fine-tune LoRA weights for CLIP using the synthetic data to improve downstream image classification over previous approaches on a large variety of datasets. We demonstrate the efficacy of DataDream through extensive experiments, surpassing state-of-the-art classification accuracy with few-shot data across 7 out of 10 datasets, while being competitive on the other 3. Additionally, we provide insights into the impact of various factors, such as the number of real-shot and generated images as well as the fine-tuning compute on model performance. The code is available at https://github.com/ExplainableML/DataDream.

3.1 Introduction

The emergence of text-to-image generative models, such as Stable Diffusion [159], not only enables us to create photo-realistic synthetic images, but it also presents opportunities to enhance downstream tasks. One potential application lies in training or fine-tuning task-specific models on synthetic data. This is shown to be particularly useful in domains where access to real data is limited [37, 69, 109, 197], as generative models offer a cost-effective means of generating large amounts of training data. In this paper, we study the impact of synthetic training data on image classification tasks in low-shot settings, i.e.

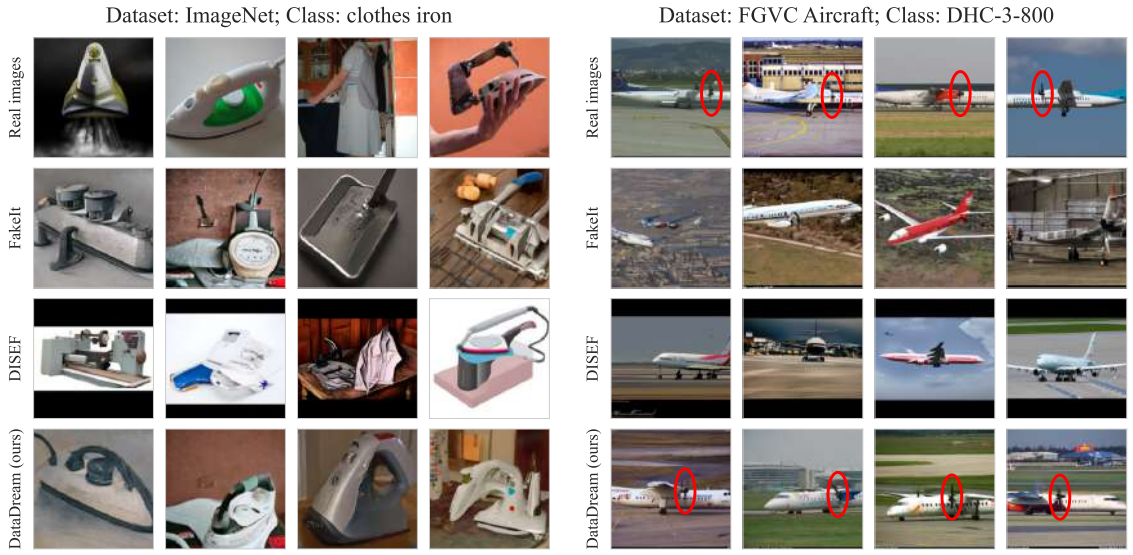


Figure 3.1: **Synthetic images comparison.** The previous methods for synthesizing training data sometimes misunderstand the class name due to its ambiguity (FakeIt [173] confuses the clothes iron with the metal iron) or fail to capture fine-grained features (DISEF [37] generated images lack the propeller in front of the wings in the DHC-3-800 aircraft, a red circle indicates the propeller). Meanwhile, our method accurately generates images of the class of interest and captures fine-grained details.

where we have access to a few images per class, but the collection of an entire dataset would be prohibitively expensive.

Previous research has primarily focused on using the class names of a given dataset [69, 173, 185, 227] to inform the data generation process. Concretely, they generated images with text-to-image diffusion models, using the class names as conditional input. To better guide the model to generate accurate depictions of the target object, they incorporated textual descriptions of each class to the prompt, sourced from language models [69, 227] or human-annotated class descriptions [173]. While intuitive, these methods lead to some generated images lacking the object of interest. For instance, while the real images for the class name `"clothes iron"` from the ImageNet [42] dataset display the appliance for ironing clothes, the images generated by FakeIt [173] mostly depict iron as the metal or arbitrary objects made thereof (Figure 3.1, left). This occurs when the generative model misunderstands class name ambiguities or rare classes. Such misalignment between the real and synthetic images limits the generated images' informational value for image classification and hinders performance gains.

To bridge the gap between real and synthetic images, real images can better inform the generative model about the characteristics of the real data distribution [8, 37, 49, 69, 244]. For instance, the concurrently developed DISEF [37] method uses few-shot samples as conditional input to the pre-trained diffusion model by starting from a partially noised real image when generating the synthetic dataset. It additionally uses a pre-trained captioning model to diversify the text-to-image prompt. While this approach improves

the alignment of real and synthetic data distributions, it sometimes falls short of capturing fine-grained features. For example, while the real images for the class name `"DHC-3-800"` in the Aircraft [121] dataset include a propeller in front of the wings, the synthetic images by DISEF lack this detail (Figure 3.1, right). Accurately representing class-discriminative features can be critical for classification tasks, particularly in fine-grained datasets.

In this work, we propose a novel approach, called **DataDream**, aimed at adapting generative models using few-shot real data. Motivated by personalized generative modeling methods [60, 168], in which generation models are fine-tuned with a small set of real images depicting an *identical object*, our method focuses on aligning the generative model to a target dataset which has *multiple classes and diverse objects for each class*. This differs from previous few-shot dataset generation methods such as [37, 69], which have not explored fine-tuning the generative model. Concretely, we adapt Stable Diffusion [159] with LoRA [80] in two ways: **DataDream$_{\text{cls}}$**, which trains LoRA per class, and **DataDream$_{\text{dset}}$**, which trains a single LoRA for all classes. To the best of our knowledge, we are the first to propose using few-shot data to adapt the generative model for synthetic training data, rather than leveraging the frozen, pre-trained generation model. Following training, we generate images with the same prompt used for fine-tuning DataDream, resulting in images depicting the object of interest (e.g. the clothes iron) or fine-grained features (e.g. the propeller of the DHC-3-800 plane) as shown in the last row of Figure 3.1.

We demonstrate the effectiveness of DataDream through extensive experiments, achieve the state of the art across all datasets when using only synthetic data, and achieve the best performance on 7 out of 10 datasets when training with both real few-shot and synthetic data. To understand the effectiveness of our method, we analyze the alignment between real and synthetic data, revealing that our method shows better alignment with the distribution of real data compared to baseline methods. Finally, we explore the scalability of our method by increasing the number of synthetic data points and real samples, showing the potential benefits of larger datasets. To summarize, the contributions of our work are as follows:

1. We introduce DataDream, a novel few-shot method which adapts Stable Diffusion to generate better in-distribution images for downstream training that outperforms state-of-the-art few-shot classification on 7 out of 10 datasets, with the other 3 comparable.

2. We emphasize the importance of reporting results with only synthetic data. We demonstrate that our method achieves superior performance when training the classifier with solely synthetic data, in some cases outperforming those trained solely with real few-shot images, indicating that our method generates images that glean more insightful information from the few-shot real data.

3. We study the effectiveness of our method by analyzing the distribution alignment between synthetic data and real data. Under few-shot guidance, synthetic data

generated by our method aligns the best with real data.

3.2 Related work

Synthetic image generation has made immense progress, now being capable of generating images that even humans may find difficult to distinguish from real images. In the following, we review related work on image generation and training on synthetic data.

Synthetic Image Generation. The suite of image generation models is growing, including Variational Auto-Encoders [94], GANs [63], and Diffusion Models [159]. With their recent popularity, diffusion models such as Stable Diffusion [159], SDXL [151], DALL-E [13, 157], Imagen [170], GLIDE [137], and Wuerstchen [149] have revolutionized text-to-image generation. Diffusion models aim to incrementally de-noise data by modeling the reverse process of a Markov chain progressively adding Gaussian noise to the sample conditioned on text. At test-time, this facilitates the generation of synthetic images from specified text and random noise. These large pre-trained models can be adapted to user specific needs [60, 168] or better control generation [33, 113]. Textual inversion [60] uses a small number of images of a specific object to learn a representational language token which can be used to prompt the frozen generation model to create better images of that object (e.g. a photo of *your* cat, rather than *a* cat). On the other hand, DreamBooth [168] achieves personalization by fine-tuning the generation model while providing a unique input token with two losses: one to reconstruct the personalized concept, and the other to preserve the original model generations without the unique token.

Training with Synthetic Data. A pool of research has blossomed in its wake, exploring downstream applications; namely: can models be trained on synthetic data? Some works augmented real datasets with synthetic images [10, 22, 49, 244]. Others focused on pre-training on large amounts of synthetic data, followed by fine-tuning on a limited number of real images [65, 197]. Similarly, several works evaluated the effectiveness of training on entirely synthetic datasets [65, 173].

Different tasks have been considered, including classification [10, 22, 69, 173, 185, 244], object detection [109], image generation [3], and representation learning [197]. Attempts have been made to optimize the selection process from large pools of synthetic data, generally by focusing on two primary factors: faithfulness and diversity. Faithfulness has been addressed by CLIP filtering [49, 69, 109], including additional class information [173], and spectral clustering [109]. On the other hand, diversity can be increased by lowering the guidance scale [173], generating a wide variety of natural language prompts with LLMs [65, 69], specifying domains [49, 185] or backgrounds [173], and using multiple text prompt templates [22]. Generally, data collection is considered resource intensive, while generating synthetic data is comparatively inexpensive and can therefore be done at scale; [173] showed that as the number of synthetic images increases, model performance can even surpass that of models trained on a lower fixed number of real images.

Finally, the few-shot setting is seeing increased interest, where the focus lies on leveraging large amounts of synthetic data in conjunction with limited amounts of real data. More than simply pooling the data sources together, we can guide the generation of better synthetic data with real data. In [69], the authors explored two strategies: 1) generating images by starting from a partially noised few-shot sample and 2) using the similarity of synthetic image features to real ones to remove low-confidence samples. When adapting the CLIP model using Classifier Tuning [215] the first strategy works best. Concurrently to our work, Diversified In-domain Synthesis with Efficient Fine-tuning (DISEF) [37] proposes to create a synthetic augmentation pipeline which leverages few-shots by starting the geration process from a noised real sample (same as [69]), then promotes diversity by denoising it conditioned on the caption from a different real image. The authors apply CLIP filtering to remove synthetic images which would be classified incorrectly and then adapt CLIP as the classifier with LoRA [80] on either the few-real shots alone or the combination of few-shots and synthetic data. In contrast to these methods, we propose to additionally fine-tune the diffusion model with LoRA to obtain a better alignment with the real data distribution.

3.3 Methodology

In this section, we start by describing the preliminaries in §3.3.1, before introducing DataDream in §3.3.2. DataDream fine-tunes the text-to-image diffusion model with few-shot data. To measure performance, synthetic images are generated with the adapted model and a classifier trained on both synthetic and real data.

3.3.1 Preliminaries

Latent diffusion model. We implement our method based on Stable Diffusion [159], a probabilistic generative model that learns to generate realistic images using a textual prompt. Given data $(x, c) \in \mathcal{D}$, where x is an image and c is a caption describing x, the model learns a conditional distribution $p(x|c)$ by gradually denoising the Gaussian noise in the latent space. Given a pretrained encoder E that encodes an image x to a latent z, i.e. $z = E(x)$, the objective function is defined as:

$$\min_{\theta} \mathbb{E}_{(x,c)\sim\mathcal{D},\, \epsilon\sim\mathcal{N}(0,1),\, t} \left[\| \epsilon - \epsilon_\theta(z_t, \tau(c), t) \|_2^2 \right] , \tag{3.1}$$

where t is a timestep, z_t is a latent noised t steps from the latent z, τ is a text encoder, and ϵ_θ is a latent diffusion model. Intuitively, the parameters θ are trained to denoise the latent z_t, given a text prompt c as conditional information. In the inference phase, a random noise vector z_T is passed through the latent diffusion model T times, along with the caption c, to get a denoised latent z_0. z_0 is then fed into a pretrained decoder D to get an image $x' = D(z_0)$ for the text-to-image generation.

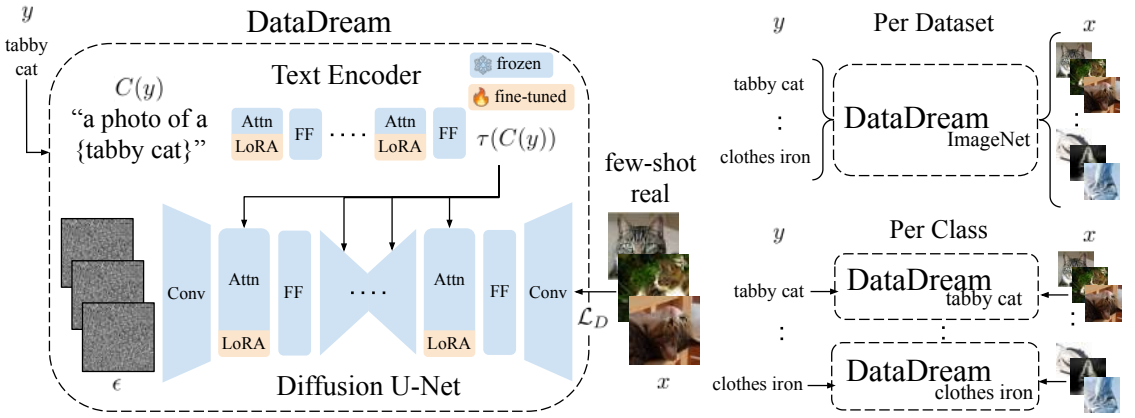


Figure 3.2: Overview of DataDream. We fine-tune LoRA weights for the linear weights of the attention layers in both the text-encoder and the diffusion U-net to generate images closer to the few-shot images. We can train one set of DataDream weights for the whole dataset sharing common dataset-specific characteristics between classes, or a separate set of weights for each class to better learn fine-grained details of each classes.

Low-rank adaptation. The Low-Rank Adaption method (LoRA) [80], is a fine-tuning method to adapt a large pre-trained model to downstream tasks in a parameter-efficient manner. Given pre-trained model weights $\theta \in \mathbb{R}^{d\times k}$, LoRA introduces a new parameter $\delta \in \mathbb{R}^{d\times k}$ that is decomposed into two matrices, $\delta = BA$, where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ with small LoRA rank r, $r \ll \min(d, k)$. The LoRA weights are added to the model weights to obtain the fine-tuned weights, i.e. $\theta^{(\text{ft})} = \theta + \delta$, for adaptation to downstream tasks. During training, θ remains fixed while only δ is updated.

3.3.2 DataDream method

Our goal is to improve classification performance by leveraging synthetic images generated by diffusion models. To this end, it is crucial to align the synthetic image distribution to that of the real images. We achieve alignment by adapting the diffusion model to a few-shot dataset of real images.

We assume access to a few-shot dataset $\mathcal{D}^{\text{fs}} = \{(x_i, y_i)\}_{i=1}^{KN}$, where x_i is an image, $y_i \in \{1, 2, \cdots, N\}$ is its label, K is the number of samples per class, and N is the number of classes. To match the real data distribution, we fine-tune it with the few-shot dataset $\mathcal{D}^{\text{fs}}$. Concretely, we introduce LoRA weights in both the text-encoder and the U-net of the diffusion model, where we make the parameter-efficient choice of adapting the attention layers. For every attention layer, we consider the query, key, value, and output projection matrices W_q, W_k, W_v, W_o, where for each matrix, the linear projection is replaced by

$$h_{l,\star} = W_\star h_{l-1} + B_\star A_\star h_{l-1} \tag{3.2}$$

with h representing the input/output activations of the projections, and resulting in the trainable LoRA weights $\delta^{(l)} = \{A_\star, B_\star | \forall \star \in \{q, k, v, o\}\}$ for every attention layer l. We omit bias weights for notational simplicity. All other model parameters are kept frozen

(including $W_\star$) while δ weights are optimized with gradient descent. To start training from the pre-trained diffusion model checkpoint, weight matrices $B_\star$ are initialized with zeros while $A_\star$ is initialized randomly. As a result, the combined fine-tuning weights $B_\star A_\star$ are zero initially and incrementally learn modifications to the original pre-trained weights. At test time, LoRA weights can be integrated into the model by updating the weights with $W_\star^{(\text{ft})} = W_\star + B_\star A_\star$, such that inference time is equivalent to the pre-trained model. In contrast to DreamBooth [168], we do not fine-tune all network weights and do not add a preservation loss, as its regularization would prevent a strong alignment with the real images.

We further consider two settings: 1) **DataDream$_{\text{dset}}$**, where we train the LoRA weights of the diffusion model on the whole dataset $\mathcal{D}^{\text{fs}}$, and 2) **DataDream$_{\text{cls}}$**, where we initialize N sets of LoRA weights $\{\delta_n | n = 1, \cdots, N\}$, one for each of the dataset classes trained on the subset $\mathcal{D}_n^{\text{fs}} = \{(x, y) | (x, y) \in \mathcal{D}^{\text{fs}}, y = n\}$.

In the DataDream$_{\text{dset}}$ setting, the original model parameters θ are kept frozen and only the LoRA weights are trained with the objective function

$$\min_{\delta} \mathcal{L}_{\text{D}} = \min_{\delta} \; \mathbb{E}_{(x,y)\sim\mathcal{D}^{\text{fs}},\, \epsilon\sim\mathcal{N}(0,1),\, t} \left[\| \epsilon - \epsilon_{\theta,\delta}(z_t, \tau_\delta(C(y)), t) \|_2^2 \right] . \tag{3.3}$$

In the DataDream$_{\text{cls}}$ setting, $\mathcal{D}_n^{\text{fs}}$ and δ_n would replace $\mathcal{D}^{\text{fs}}$ and δ, respectively. Since we use a text-to-image diffusion model, we define the text condition through the function C which maps the label y, i.e. class name, to a prompt using the standard template, `"a photo of a [CLS]"` [155, 243]. The prompt is passed through the text encoder and subsequently used during the decoding steps of the diffusion model. We illustrate both DataDream fine-tuning and our two settings in Figure 3.2.

Both settings have distinct advantages. In DataDream$_{\text{dset}}$, LoRA weight sharing between classes allows knowledge transfer about common characteristics within the whole dataset. This would be beneficial in a fine-grained dataset that shares the coarse-grained features across classes. On the other hand, DataDream$_{\text{cls}}$ allocates more weights to learn about details of each class, which allows the generation model to better align with the per-class data distribution.

After adapting the diffusion model to the few-shot dataset, we generate 500 images per class with the adapted model conditioned on the same textual prompt used for DataDream, forming a synthetic dataset $\mathcal{D}^{synth}$. We train a classifier on either only synthetic images or the combination of synthetic and real few-shot images $\mathcal{D}^{fs}$.

For classifier training, we adapt a CLIP model [155], similar to previous work in few-shot classification [37]. We add LoRA adaptors [80] to both image encoder and text encoder of CLIP ViT-B/16 model [155]. When training with synthetic and real images jointly, we use a weighted average of the losses from real data and synthetic data,

$$\mathcal{L}_{\text{C}} = \; \lambda\, \mathbb{E}_{(x,y)\sim\mathcal{D}^{\text{fs}}} \text{CE}(f(x), y) + (1-\lambda)\, \mathbb{E}_{(x,y)\sim\mathcal{D}^{\text{synth}}} \text{CE}(f(x), y) \,, \tag{3.4}$$

where λ is the weight assigned to the loss from real data and the function CE is a cross-entropy loss.

3.4 Experiments

In this section, we present our experimental results on DataDream. We present details of the experimental setup in §3.4.1. In §3.4.2, we compare our methods to baselines both quantitatively and qualitatively. Furthermore, we analyze the synthetic data of DataDream to understand why it outperforms baselines in §3.4.3, followed by ablation studies in §3.4.4.

3.4.1 Experimental setup

Benchmarks. We evaluate our method on 10 datasets: ImageNet [42], Oxford Pets [145] containing fine-grained pet classes, FGVC Aircraft [121] containing fine-grained aircraft classes, Food101 [17] containing common food classes, Stanford Cars [100] containing fine-grained car classes, DTD [36] with texture images, EuroSAT [72] with satellite images, Flowers 102 [139] containing fine-grained flower classes, SUN397 [218] with scene images, and Caltech 101 [105] with pictures of common objects.

Implementation details. We implement DataDream based on Stable Diffusion [159] version 2.1. For each seed, we randomly sample the few-shot images from the training samples of each dataset. Our method is trained for 200 epochs with a batch size of 8 for all datasets, with the exception of $\text{DataDream}_{\text{dset}}$ on ImageNet, which is trained for 100 epochs. Hence, $\text{DataDream}_{\text{dset}}$ and $\text{DataDream}_{\text{cls}}$ share the same amount of training compute, i.e. each of the N $\text{DataDream}_{\text{cls}}$ adapter weights (one for each class) performs S/N update steps where S is the total number of steps of $\text{DataDream}_{\text{dset}}$ for the whole dataset. We use AdamW [118] as an optimizer and learning rate $1e-4$, with a cosine annealing scheduler. We use LoRA rank $r=16$ for all adapted weights in DataDream. For synthetic image generation with DataDream, we use 50 steps and guidance scale 2.0. We generate 500 images per class if not mentioned otherwise. For the classifier, we use CLIP ViT-B/16 [155] as a base model, and fine-tune LoRA applied on both the image encoder and text encoder of CLIP with rank 16. We set the weight assigned to the real loss term to $\lambda=0.8$. DataDream is computed on three random seeds. Additional implementation details can be found in Appendix B.

Baseline methods. As all methods adapt CLIP ViT-B/16 as the classifier, we provide CLIP zero-shot performance as a baseline. In our first setting, we update the classifier using only synthetic data. For this, we compare against two alternative data-generation methods: IsSynth [69] and DISEF [37]. In our second setting, classifier adaptation uses the synthetic data in addition to the few-shot real data. We refer to LoRA [80] training with only the real few-shot data as Real-finetune, to signify that this is the baseline version of DataDream, without the benefit of our synthetic data, as done in [37]. $\text{DataDream}_{\text{dset}}$, $\text{DataDream}_{\text{cls}}$, and DISEF [37] all build upon this foundation. These experiments highlight the benefit of adding synthetic data to the real few-shot images. We also compare against several

Method	R	S	IN	CAL	DTD	EuSAT	AirC	Pets	Cars	SUN	Food	FLO	Avg
CLIP (zero-shot) [155]			70.2	96.1	46.1	38.1	23.8	91.0	63.1	63.8	85.1	71.8	64.1
IsSynth [69]		✓	$70.0_{\pm 0.6}$	$95.7_{\pm 0.7}$	$67.6_{\pm 0.5}$	$71.3_{\pm 2.9}$	$34.5_{\pm 3.4}$	$92.1_{\pm 0.6}$	$65.9_{\pm 0.0}$	$72.2_{\pm 0.0}$	$85.4_{\pm 0.1}$	$90.0_{\pm 0.2}$	$74.5_{\pm 0.9}$
DISEF [37]		✓	$67.1_{\pm 0.2}$	$93.4_{\pm 1.0}$	$66.1_{\pm 0.6}$	$69.2_{\pm 2.7}$	$26.8_{\pm 2.2}$	$91.0_{\pm 0.0}$	$63.2_{\pm 0.0}$	$73.5_{\pm 0.1}$	$85.1_{\pm 0.0}$	$85.4_{\pm 0.7}$	$72.1_{\pm 0.8}$
$\text{DataDream}_{\text{cls}}$ (ours)		✓	$\mathbf{71.6}_{\pm 0.2}$	$\mathbf{96.4}_{\pm 0.0}$	$68.6_{\pm 2.0}$	$\mathbf{85.4}_{\pm 2.9}$	$60.3_{\pm 0.9}$	$\mathbf{94.2}_{\pm 0.4}$	$90.5_{\pm 0.3}$	$74.5_{\pm 0.0}$	$\mathbf{86.9}_{\pm 0.1}$	$97.2_{\pm 0.2}$	$82.6_{\pm 0.7}$
$\text{DataDream}_{\text{dset}}$ (ours)		✓	$71.5_{\pm 0.0}$	$96.2_{\pm 0.1}$	$\mathbf{69.5}_{\pm 1.2}$	$80.3_{\pm 4.1}$	$\mathbf{71.2}_{\pm 0.1}$	$94.0_{\pm 0.1}$	$\mathbf{92.2}_{\pm 0.1}$	$\mathbf{74.5}_{\pm 0.1}$	$86.7_{\pm 0.1}$	$\mathbf{98.0}_{\pm 0.4}$	$\mathbf{83.4}_{\pm 0.7}$
Real-finetune	✓		$73.4_{\pm 0.2}$	$96.8_{\pm 0.1}$	$78.3_{\pm 2.8}$	$93.5_{\pm 0.7}$	$59.3_{\pm 2.8}$	$94.0_{\pm 0.1}$	$87.5_{\pm 0.6}$	$77.1_{\pm 0.1}$	$\mathbf{87.6}_{\pm 0.0}$	$98.7_{\pm 0.1}$	$84.6_{\pm 0.8}$
IsSynth [69]	✓	✓	$73.9_{\pm 0.1}$	$97.4_{\pm 0.2}$	$\mathbf{81.6}_{\pm 0.4}$	$93.9_{\pm 0.1}$	$64.8_{\pm 0.8}$	$92.1_{\pm 0.1}$	$88.5_{\pm 0.3}$	$\mathbf{77.7}_{\pm 0.0}$	$86.0_{\pm 0.0}$	$99.0_{\pm 0.0}$	$85.5_{\pm 0.2}$
DISEF [37]	✓	✓	$73.8_{\pm 0.2}$	$97.0_{\pm 0.1}$	$81.5_{\pm 0.6}$	$\mathbf{94.0}_{\pm 0.5}$	$64.3_{\pm 0.4}$	$92.6_{\pm 1.2}$	$87.9_{\pm 0.5}$	$77.6_{\pm 0.1}$	$86.2_{\pm 0.6}$	$99.0_{\pm 0.2}$	$85.4_{\pm 0.4}$
$\text{DataDream}_{\text{cls}}$ (ours)	✓	✓	$73.8_{\pm 0.1}$	$\mathbf{97.6}_{\pm 0.2}$	$\mathbf{81.6}_{\pm 0.4}$	$93.8_{\pm 0.3}$	$68.3_{\pm 0.4}$	$94.5_{\pm 0.3}$	$91.2_{\pm 0.2}$	$77.5_{\pm 0.1}$	$87.5_{\pm 0.1}$	$\mathbf{99.4}_{\pm 0.2}$	$86.5_{\pm 0.4}$
$\text{DataDream}_{\text{dset}}$ (ours)	✓	✓	$\mathbf{74.1}_{\pm 0.3}$	$96.9_{\pm 0.7}$	$\mathbf{81.6}_{\pm 0.6}$	$93.4_{\pm 0.0}$	$\mathbf{72.3}_{\pm 0.2}$	$\mathbf{94.8}_{\pm 0.3}$	$\mathbf{92.4}_{\pm 0.1}$	$77.5_{\pm 0.1}$	$\mathbf{87.6}_{\pm 0.1}$	$\mathbf{99.4}_{\pm 0.1}$	$\mathbf{87.0}_{\pm 0.4}$

Table 3.1: **Few-shot classification performance with DataDream using real 16-shot and synthetic images** where the training dataset includes synthetic data only (top), or synthetic data + 16 real shots (bottom). All results use CLIP ViT-B/16 as the base classification model, and 500 synthetic images generated by 16 real shots. Datasets are IN: ImageNet, CAL: Caltech 101, EuSAT: EuroSAT, AirC: FGVC Aircraft, FLO: Flowers 102. R/S means using real/synthetic images for fine-tuning. DataDream and baseline methods are computed on three random seeds.

SOTA few-shot methods. In these, we include two Parameter Efficient Fine-Tuning (PEFT) techniques: VPT [85] and CoOp [243], which only use real few-shot data. We additionally compare to two SOTA image generation techniques, IsSynth [69] and DISEF [37]. For fair comparisons, we use Stable Diffusion v2.1 to generate images for all baselines instead of the originally used GLIDE [137] or Stable Diffusion v1.5. More details are described in Appendix C.

3.4.2 Classification performance with DataDream

Quantitative results on solely synthetic data. We refer to the upper portion of Table 3.1 for the synthetic-only setting, where we show that DataDream-generated data achieves state-of-the-art results on all 10 datasets. For example on FGVC Aircraft [121], $\text{DataDream}_{\text{dset}}$ achieves an impressive 47.4% point increase over the CLIP zero-shot model. In addition, on Stanford Cars [100] $\text{DataDream}_{\text{dset}}$ achieves 92.2%, while IsSynth [69] is at 65.9% and DISEF [37] at 63.2%. On Flowers102 [139], $\text{DataDream}_{\text{dset}}$ obtains 98.0% while IsSynth and DISEF reach only 90.0% and 85.4%, respectively. These boosts signify that DataDream is able to closely follow the real few-shot data distribution in its generated images.

We believe that this evaluation benchmark allows the best assessment of the quality of the synthetic image generations for training image classifiers. While adding real data to the synthetic images at training time generally provides a performance boost, it also makes it harder to quantify the quality of the synthetic images for the task, because most of the improvement still stems from the real images. Hence, issues with synthetic data generation, such as redundancy or class misrepresentation, will be more visible in synthetic-only benchmarks.

Method	S	IN	CAL	DTD	EuSAT	AirC	Pets	Cars	SUN	Food	FLO	Avg
VPT [85]		69.6	95.4	66.1	92.3	36.2	91.8	69.0	70.5	87.0	91.0	76.9
CoOp [243]		68.0	95.2	70.7	87.1	45.5	89.9	81.4	73.0	83.7	97.6	79.2
IsSynth [69]	✓	73.9	**97.4**	**81.6**	93.9	64.8	92.1	88.5	**77.7**	86.0	99.0	85.5
DISEF [37]	✓	73.8	97.0	81.5	**94.0**	64.3	92.6	87.9	77.6	86.2	99.0	85.4
$\text{DataDream}_{\text{dset}}$ (ours)	✓	**74.1**	96.9	**81.6**	93.4	**72.3**	**94.8**	**92.4**	77.5	**87.6**	**99.4**	**87.0**

Table 3.2: **Comparing DataDream with few-shot SOTA.** We compare DataDream with SOTA few-shot methods. The base setting, dataset abbreviations, and setting notations match those in Table 3.1. S indicates methods using synthetic data generation.

Comparing $\text{DataDream}_{\text{cls}}$ and $\text{DataDream}_{\text{dset}}$, the results are split over method superiority. We hypothesize that this difference comes from inherent dataset properties. For datasets where all classes share high visual similarity, sharing weights becomes beneficial as distribution characteristics generalize across classes. For example, we find that FGVC Aircraft [121] and Stanford Cars [100] show a significant advantage of $\text{DataDream}_{\text{dset}}$ over $\text{DataDream}_{\text{cls}}$. On the other hand, datasets where classes span a wide range benefit from fully specializing to the unique classes, as seen in the results for Caltech101 [105] and Food101 [17].

Quantitative results on real + synthetic data. In Table 3.1 (bottom), we present the results for the synthetic + real setting. Real-finetune provides the foundation for this section, consisting of LoRA applied to CLIP with the real few-shot data. DISEF, $\text{DataDream}_{\text{cls}}$, and $\text{DataDream}_{\text{dset}}$ build upon this by adding their respective synthetic images to the training data, which is generated from the same few-shot data. Comparing $\text{DataDream}_{\text{dset}}$ and $\text{DataDream}_{\text{cls}}$ to Real-finetune, we observe that our synthetic data improves performance on 9 out of 10 datasets over naive use of real few-shot data. For example, on FGVC Aircraft [121], our synthetic data facilitates an improvement of 13.0% over using the real few-shot data naively. In comparison, DISEF only achieves a 5.0% increase. Furthermore, we improved upon Stanford Cars [100] by 4.9%, where DISEF saw only a 0.4% increase. This shows that generating images with DataDream consistently provides value not only over naive use of the few-shot examples, but also over other data generation techniques. In fact, especially in case of the Stanford Cars dataset, the real images do not provide more information than the synthetic images generated by our model (92.2% on synthetic only vs 92.4% real + synthetic settings), which is an exciting observation.

Quantitative results comparing with SOTA. We compare DataDream with SOTA few-shot methods in Table 3.2. Ours, i.e. $\text{DataDream}_{\text{dset}}$, improves over the previous SOTA on 7 out of 10 datasets, while being competitive on the other 3. On the FGVC Aircraft [121] dataset, we improve SOTA by 7.5%, on Pets [145] by 2.2%, and on Stanford Cars [100]

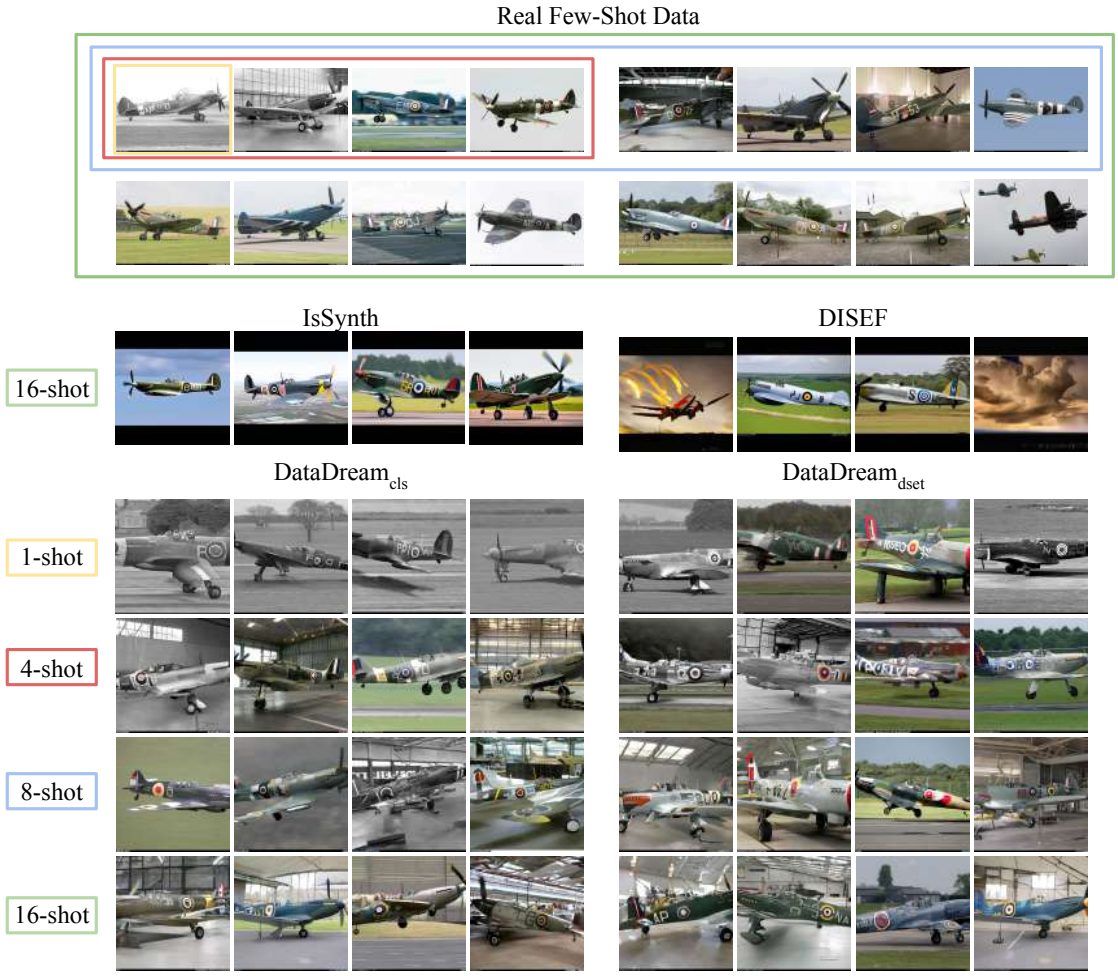


Figure 3.3: **Qualitative results with increasing number of shots vs 16-shot images generated with SOTA** of the class Spitfire from the FGVC Aircraft [121] dataset. The real few-shot images at the top are used to generate the presented synthetic images at the bottom. We always use a fixed set of 16 samples, i.e. 1-shot image is a subset of 16-shots, to insure fairness in comparing results with the increasing number of shots.

by 3.9%. This highlights that synthetic images generated by DataDream provide more training value than the previous SOTA generation method. On average, we improve over the next best synthetic augmentation method by 1.5% and over the best method without data generation, CoOp [243], by 7.8%.

3.4.3 Analysis of DataDream

Qualitative results. We provide a qualitative analysis of DataDream in Figure 3.3 of the Spitfire class in FGVC Aircraft. To support our 1-, 4-, 8-, and 16-shot generated images, we include the real few-shot examples used to generate them. We also show previous SOTA images for comparison, from two other image-generation methods: DISEF [37] and IsSynth [69].

When comparing to the previous SOTA, we notice that DataDream is better able to

generate images that match the target domain. For example, they imitate that in the context of the dataset, planes are more likely to be photographed on the ground, in a hangar, or taking off than in the air, unlike both previous methods that generate images that are unlikely to be found in the dataset. We also notice that our models are better able to match the color palette of the real data, as opposed to DISEF, where we find the colors to be too bright compared to the real data. Furthermore, our model is the only one of all three that replicates the black border at the bottom of all images, even after only a single shot.

Furthermore, we notice that DISEF has a higher tendency to generate out-of-distribution data, sometimes omitting the target class entirely and therefore creating a need for CLIP filtering. We hypothesize this might be due to their use of diverse captions, which may sometimes guide the image generation too far from the core distribution. By focusing on fidelity to the class distribution and keeping our prompts simple, we generate fewer out-of-distribution samples. This allows us to use all samples generated, which is a better use of resources.

We also notice some interesting differences between $\text{DataDream}_{\text{dset}}$ and $\text{DataDream}_{\text{cls}}$. On the one hand, we find that $\text{DataDream}_{\text{cls}}$ is better able to accurately represent the Spitfire class, especially at a low number of shots. On the other hand, the additional data in $\text{DataDream}_{\text{dset}}$ allows it to avoid certain overfitting mistakes, such as creating only black and white images after the first shot, which happens to be a monochrome image.

Comparing DataDream models trained on different numbers of few-shot examples, we notice an increase in quality with number of images, showing qualitatively the benefit of adding even a few more images. Already at four shots, we obtain images that are not only better quality, but closer to the real data domain. We also note that the lower the number of shots, the more the model benefits from careful selection of a diverse and representative group, so that the model does not pick up on patterns that are not representative of the full data distribution. At only four shots, we notice that the real images contain a specific color palette that is not necessarily representative of the full dataset, as evidenced by the next four images; this led to the 4-shot results lacking diversity of color and brightness. This goes to show that wherever possible, careful selection of representative data is highly beneficial. It also highlights the ability of our method to find and replicate patterns in the few-shot distribution.

Distribution alignment. The qualitative analysis shows DataDream being able to capture both the presence of objects of interest and the fine-grained features essential for class discrimination. To gain more insights, we examine the alignment between synthetic and real datasets. To quantitatively assess the alignment, we use the Frechet Inception Distance (FID) [76] score, a metric that quantifies the quality of generated images. Concretely, we compute the set of FID scores for each method by evaluating the distance between the distribution of synthetic images and that of real images on a per-class basis. Lower FID

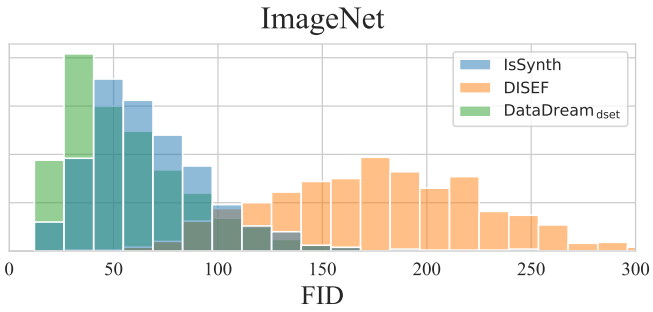


Figure 3.4: **Distribution of FID scores per-class.** The FID score is calculated per-class to measure how close the synthetic data distribution is to the real data distribution.

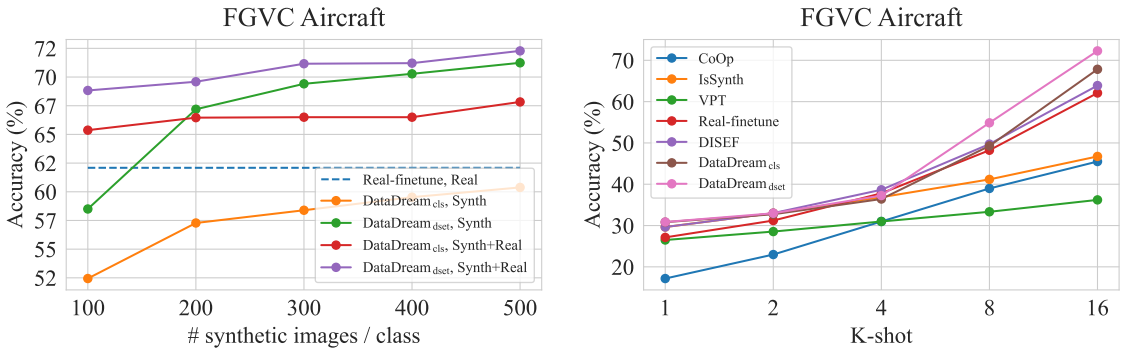


Figure 3.5: **Ablation study of DataDream.** Left: We vary the number of synthetic images per class to understand the scaling effect. Right: We vary the number of real examples used for training DataDream.

indicates synthetic images are closer to the real data distribution. We visualize the FID scores using histograms, as shown in Figure 3.4.

In the experiment on ImageNet, we observe that the histogram for our method skews left, indicating lower FID scores. Meanwhile, the histogram of DISEF tends to lean towards the right. This is attributed to how DISEF uses LLM-generated prompts for image generation. While it gives the generated images diversity, it also introduces out-of-distribution artifacts, as observed in the qualitative analysis. Compared to DISEF, IsSynth aligns more with the real data, but may have less diversity due to its usage of a standard prompt, generating similar images from the conditioned real image. In contrast, our DataDream$_{\text{dset}}$ balances fidelity, due to the adaptation of the generative model to match few real shots, and diversity, due to the initial randomness in the generation pipeline. This results in the synthetic images of DataDream$_{\text{dset}}$ closely matching the real data distribution. We posit that the better alignment contributes to classification performance, as demonstrated in §3.4.2.

3.4.4 Ablation study

Accuracy scaling by number of synthetic images. Previous literature has shown that as the number of synthetic images increases, model accuracy may also increase [173].

Therefore, we provide Figure 3.5, where the left part shows the effect on DataDream of increasing the number of images for FGVC Aircraft [121]. We find that as more images are generated, model accuracy increases in all settings: DataDream$_{\text{dset}}$ and DataDream$_{\text{cls}}$ and Synth and Synth + Real. Even at 500 images, we observe that performance is not yet saturated. Compared to Real-finetune, we observe that in the Synth + Real setting, DataDream performs better already starting at 100 images per class.

In the synthetic-only setting, DataDream$_{\text{cls}}$ out-performs real images entirely after generating only 200 images. Another interesting result is the gap between DataDream$_{\text{cls}}$ and DataDream$_{\text{dset}}$: this difference holds between any number of images, showing that DataDream$_{\text{dset}}$ is a better fit for this dataset than DataDream$_{\text{cls}}$, regardless of the number of synthetic images. However, remembering from Table 3.1 that DataDream$_{\text{dset}}$ performed better than DataDream$_{\text{cls}}$ on 5 out of 10 datasets, we note that this advantage is dataset-dependent rather than a general trend.

Varying the number of few-shot images K. Furthermore, we operate in a K-shot regime; therefore, we expect that as K increases, the model accuracy should increase as well. As done in [37], we show the effect of 1-, 2-, 4-, 8-, and 16-shots on FGVC Aircraft [121] dataset, for 500 images in the real + synth setting, compared to previous literature. We observe that as the number of few-shot images increases, DataDream consistently shows higher accuracy than previous SOTA. We believe that this behavior is expected; as with any training or fine-tuning regime, at least a small training dataset base is necessary. Too few samples could be prone to overfitting, thus reducing variety and failing to include enough information to successfully understand the overall class distribution. At the same, since we use LoRA on a subset of all model parameters, we limit the amount of overfitting from our fine-tuning as compared to full model fine-tuning. This allows DataDream to outperform Real-finetune even when we use only a single shot. As more data becomes available, however, DataDream is able to successfully leverage even as few as four or eight shots to noticeably adapt to the data distribution, as was shown in Section 3.4.3. Hence, we obtain a relative performance boost when compared to other methods.

3.5 Conclusion

In this paper, we studied the efficacy of leveraging the generative models for improving the image classification performance in few-shot scenarios. We proposed DataDream, a method to generate synthetic data with the guidance of few-shot samples, which are then used for training the image classifier. We introduced LoRA adaptors on both the text encoder and the diffusion U-Net to efficiently fine-tune the generative model. We proposed two variants: DataDream$_{\text{dset}}$, which trains LoRA on the whole targeted dataset, and DataDream$_{\text{cls}}$, which adapts LoRA per-class. Our experiments demonstrate that our method consistently improves classification performance across benchmarks, both in the synthetic-only and synthetic+real settings. Through qualitative analysis, we observed

that images generated by our method more precisely generate objects of interest as well as fine-grained details, contributing to their alignment with real data distributions, as quantitatively examined by FID scores. Furthermore, we investigated the scalability of our method by increasing the number of synthetic data samples and the number of real samples.

4

LoFT: LoRA-fused Dataset Generation with Few-shot Guidance

Despite recent advances in text-to-image generation, using synthetically generated data seldom brings a significant boost in performance for supervised learning. Oftentimes, synthetic datasets do not faithfully recreate the data distribution of real data, i.e., they lack the fidelity or diversity needed for effective downstream model training. While previous work has employed few-shot guidance to address this issue, existing methods still fail to capture and generate features unique to specific real images. In this paper, we introduce a novel dataset generation framework named LoFT, **Lo**RA-**F**used **T**raining-data Generation with Few-shot Guidance. Our method fine-tunes LoRA weights on individual real images and fuses them at inference time, producing synthetic images that combine the features of real images for improved diversity and fidelity of generated data. We evaluate the synthetic data produced by LoFT on 10 datasets, using 8 to 64 real images per class as guidance and scaling up to 1000 images per class. Our experiments show that training on LoFT-generated data consistently outperforms other synthetic dataset methods, significantly increasing accuracy as the dataset size increases. Additionally, our analysis demonstrates that LoFT generates datasets with high fidelity and sufficient diversity, which contribute to the performance improvement. The code is available at https://github.com/ExplainableML/LoFT.

4.1 Introduction

Synthetic data offers a cost-effective alternative to the labor-intensive process of real data collection. One promising downstream application of diffusion-based text-to-image generative models [13, 77, 151, 157, 159, 170] is to augment real datasets with synthetic images [49, 185] or training models on entirely synthetic data [69, 173, 227]. While these methods show potential, models trained solely on synthetic data often underperform compared to those trained on real data [55]. This is largely due to distributional misalignment between synthetic and real data, as well as a lack of fine-grained detail in the generated

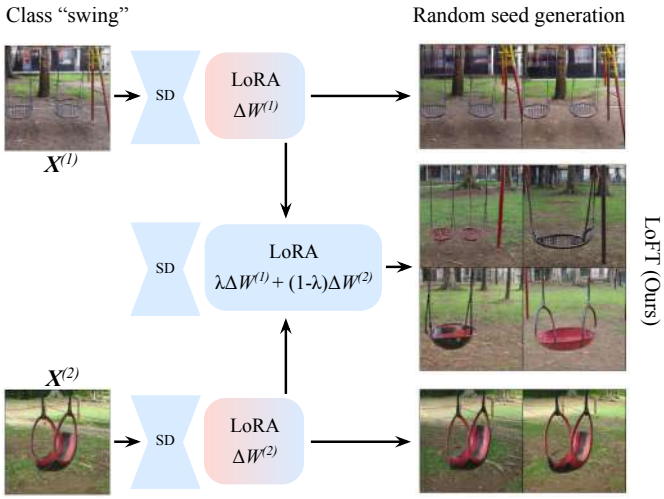


Figure 4.1: LoFT: Given a few real images per class, we first adapt a diffusion model to each image using LoRA. Next, two LoRA weights corresponding to images of the same class are randomly selected and fused to generate new images. The generated synthetic images above show diverse colors and compositions while maintaining the swing object.

images [55, 90].

To tackle distribution shift issue, recent works suggests to guide the dataset generation with a few real data samples [37, 69, 90]. In this few-shot setting, we assume to have access to a few real images for every class for an image classification task. Yet, the issue of misalignment remains a challenge for certain classes or downstream datasets. For instance, [37] use partially noised real image as conditional input to the diffusion model and generate synthetic data using prompts from a captioning model. However, these images often deviate from the real data distribution, making them less relevant to the task at hand. [90] propose DataDream, which fine-tunes the diffusion model with few-shot data to learn the data distribution, but we find that it struggles to generate in-distribution images for all classes consistently. This is because DataDream finetunes on all available images from the same class, which makes it challenging to retain high-fidelity details of individual images (e.g., less frequently visible parts of a class, such as the back of a car), focusing instead on commonly shared features.

We introduce LoFT, **Lo**RA-**F**used **T**raining-data Generation with Few-shot Guidance to generate high-fidelity, in-distribution synthetic images using few-shot real images. Instead of fine-tuning a diffusion model on all image of a class jointly, we train separate sets of Low-Rank Adaptation (LoRA) parameters on individual images, i.e., the diffusion model learns to overfit to a single image, generating it exclusively. At inference time, we then fuse together the LoRA weights of any two real images from the same class, to generate synthetic images that share the characteristics of both images. As shown in Figure 4.1, given two images of swings, the individual LoRA weights lead to generations similar to the real image (top and bottom), while the fused LoRA weights create images inheriting features from both source images while still maintaining the identity of a swing. There

are two advantages of our LoFT method. First, learning separate LoRA weights for each individual image eases the diffusion model adaptation as the finetuning can retain on every detail of the real image. This instance-level adaptation ensures better alignment between the distribution of the few-shot real images and the synthetic images generated by the LoRA-tuned diffusion model, resulting in high fidelity. Second, by fusing the LoRA weights from different images of the same class, we maintain the diversity of the generated synthetic images.

Our key contributions are: (1) introducing LoFT, a few-shot guided synthetic dataset generation method that generates high-fidelity, in-distribution synthetic datasets by training LoRA adapters per image and fusing them when generating synthetic images; (2) providing a comprehensive comparison of four synthetic dataset generation methods on ten downstream datasets, demonstrating superior performance in fine-tuning CLIP when trained on data from LoFT; and (3) analyzing synthetic data generation methods based on fidelity and diversity, showing that LoFT achieves high fidelity with sufficient diversity, leading to improved performance when using its synthetic dataset.

4.2 Related Work

Diffusion-based text-to-image (T2I) models have enabled the creation of highly realistic synthetic images [13, 137, 151, 157, 159, 170]. These models operate by gradually denoising Gaussian noise, conditioned on textual prompts. Promising downstream applications of T2I generative models include generating training data for classification [8, 49, 59, 69, 90, 102, 173, 185, 193, 222, 227, 228, 237, 240, 244], handling long-tail distributions [15, 73, 183], data distribution shifts [10, 46], semi-supervised learning [224], representation learning [197], object detection [109], vision-language pre-training [65, 178, 196], and image generation [3, 12]. For image classification, a lot of work has focused on generating synthetic images zero-shot, which typically involves generating data from text prompts that include the class names from the downstream task [69, 173, 185, 227, 237]. However, this approach often leads to generated images that lack a faithful representation of the target object, resulting in a mismatch between synthetic and real images, which hinders performance gains [90, 196]. To mitigate these issues, there has been growing interest in few-shot learning, where limited real data is used alongside synthetic data. Techniques such as initializing the generation from a partially noised real image [37, 69] or fine-tuning the diffusion model with few-shot data [90] have been employed to align the synthetic data more closely with real-world distributions. Our method differs from other few-shot guided methods by using LoRA fusion to combine the features of multiple real images.

Controllable text-to-image diffusion models have enabled personalization in image generation [60, 154, 168, 169, 182, 203]. Fine-tuning a diffusion model with LoRA [80] and fusing them in the image generation phase has been demonstrated to be an effective technique for image morphing [236] and model customization [45], where LoRA weights are fused to achieve customized outputs. In contrast, our method leverages LoRA fusion for

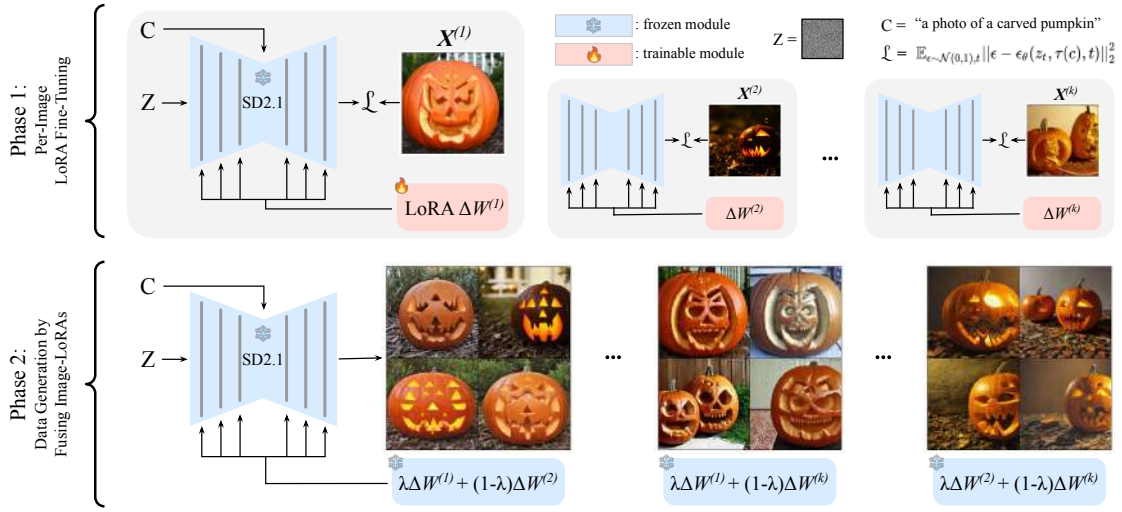


Figure 4.2: LoFT pipeline. In the first phase, given a few real images per class, we adapt a diffusion model to each image using LoRA. In the second phase, two LoRA weights corresponding to images of the same class are randomly selected and fused to generate new synthetic images. These generated images are then compiled to form a dataset for training the classification model.

synthetic dataset generation and demonstrates its effectiveness in training classification models. While [244] proposed fusing learned tokens from Textual Inversion [60] for synthetic dataset generation, we show that our LoFT outperforms these previous fusion methods.

To understand and improve the impact of synthetic training data, [55] measure the fidelity and diversity of synthetic datasets. Fidelity can be improved through methods like CLIP filtering [49, 69, 109] and incorporating additional class information [173], while diversity is affected by the guidance scale [55, 173], adding attributes to prompts [49, 173, 185], or using large language models to generate more varied prompts [65, 69]. Additionally, [55] have explored scaling laws in synthetic training data, demonstrating that synthetic datasets do not exhibit the same scaling benefits as real data in supervised tasks. In this work, we examine few-shot guided dataset generation methods in terms of both fidelity and diversity, and further investigate how scaling up synthetic datasets to sizes of up to one million affects the performance of these methods.

4.3 LoRA-Fused Training Dataset Generation

In this section, we begin by describing baseline methods for synthetic dataset generation in the zero-shot and few-shot scenarios (§4.3.1). We then introduce our proposed method, LoFT, in §4.3.2.

4.3.1 Synthetic dataset generation

Stable diffusion: text-to-image generation. The Stable Diffusion [159] model learns a conditional probability distribution $p(x|c)$ given a data point $(x, c) \in \mathcal{D}$ where x is an image and c is its caption. The model learns a reverse process of gradually denoising Gaussian noise in the latent space. Concretely, the diffusion and reverse processes work in a latent space, which is defined through a pre-trained image encoder f that encodes the image x to a latent z, i.e. $z = f(x)$, and the corresponding decoder g where $x = g(z)$. Given a time step $t \in \{0, \cdots, T\}$, z_t denotes noisy latent state after t steps of small Gaussian noise addition from $z_0 = z$ where z_T is Gaussian noise. The latent diffusion models' objective is to minimize the following loss:

$$\min_{\theta} \mathbb{E}_{(x,c)\sim\mathcal{D},\, \epsilon\sim\mathcal{N}(0,1),\, t} \left[\| \epsilon - \epsilon_\theta(z_t, \tau(c), t) \|_2^2 \right], \tag{4.1}$$

where $\tau(\cdot)$ is a text encoder. Intuitively, the loss enables the model to learn to denoise the latent z_t. During inference, we start with noise z_T and iteratively denoise it through T steps of the latent diffusion model, obtaining z_0. This latent is decoded by the pre-trained decoder g to generate a final image $x' = g(z_0)$.

Zero-shot image generation. New images can be generated by conditioning the model on a template prompt [69, 173], such as `"a photo of a {`l`}"`, where l represents a class name. As a result, the synthetic dataset

$$\mathcal{D}^{\text{synth}} = \{(x_i, y_i)\}_{i=1}^{sL}$$

contains s generated images for each of the L classes where every image x_i is automatically annotated by the class label $y_i \in \{1, 2, \cdots, L\}$ derived from its textual prompt. To improve diversity in these generated images, lowering the guidance scale has been shown to be effective, as it encourages more output variety, therefore improving classification performance [55, 173]. We refer to this method as **ClassPrompt**.

Few-shot guided dataset generation. While zero-shot text-to-image methods can generate a large amount of distinct images, they often struggle to produce the classification object of interest or capture fine-grained details of a class [90]. To address this, few-shot guided approaches have been developed, where we assume access to a few real images for each class. In the k-shot setting, we denote

$$\mathcal{D}^{\text{fs}} = \{(x_i, y_i)\}_{i=1}^{kC}$$

as the few-shot dataset, where x_i is an image, y_i is the label of the image, k is the number of available real images per class, and C is the number of classes. Few images per class can already provide rich visual information to better inform the data generation process beyond textual labels alone, while not requiring an extensive effort to collect. For instance, DataDream [90] fine-tunes LoRA weights applied on the diffusion model using few-shot

real images. In our experiments, we use DataDream with LoRA weights trained for each class as a representative of the few-shot guided image generation approach based on fine-tuning, and refer to this as **DataDream**.

While lowering the guidance scale increases variety in zero-shot image generation, using a template text prompt still limits the generation of a diverse dataset. To further increase diversity, [227] have leveraged large language models (LLMs) to enrich prompts with additional context or attributes related to the class name. Additionally, [49] and [55] leverage real images by applying a captioning model to create detailed captions from these images, which are then used as prompts for generation. In our experiments, we include a baseline for few-shot guided data generation through captioning. Specifically, we caption the few-shot images for each class with PaliGemma [14], a multimodal large language model. We generate one caption per real image, i.e., k captions per class in the k-shot setting. We then use these captions as prompts for synthetic image generation. We refer to this method as **CaptionPrompt**.

4.3.2 LoFT method

While DataDream has shown promising results for few-shot guided dataset generation, we find that it struggles to generate in-distribution images for some classes, limiting its impact on classification performance. This issue arises when there is high diversity in the few-shot images such that the fine-tuned diffusion models do not faithfully represent fine-grained details that may occur only in one of the images, leading to underfitting.

To overcome these challenges, we propose LoFT, **Lo**RA-**F**used **T**raining-data Generation with Few-shot Guidance for generating better in-distribution synthetic images. As shown in Figure 4.2, LoFT fine-tunes the pre-trained diffusion model with one set of LoRA weights for *every real image* x_i from the few-shot dataset $\mathcal{D}^{\text{fs}}$ independently. Specifically, for every attention layer of the diffusion model U-net, we add LoRA [80] parameters to the linear weight matrices

$$h_{\text{out}} = W h_{\text{in}} + \Delta W^{(i)} h_{\text{in}} \tag{4.2}$$

where h is the activation of a linear layer, $W \in \mathbb{R}^{d_1 \times d_2}$ is the original weight matrix, and $\Delta W^{(i)}$ is the low rank adaptation matrix which is optimized. The parameterization

$$\Delta W^{(i)} = B^{(i)} A^{(i)}$$

with $B \in \mathbb{R}^{d_1 \times r}$ and $A \in \mathbb{R}^{r \times d_2}$ allows for efficient fine-tuning because the low rank $r \ll min(d_1, d_2)$ reduces the number of tunable parameters significantly. The parameter efficiency and modularity of LoRA lead to a low storage costs and flexible inference-time manipulation (fusion).

Fine-tuning LoRA weights on a single image. As presented in the grey box in Figure 4.2, we fine-tune a separate set of LoRA weights $\Delta W^{(i)}$ for each data point (x_i, y_i) with the

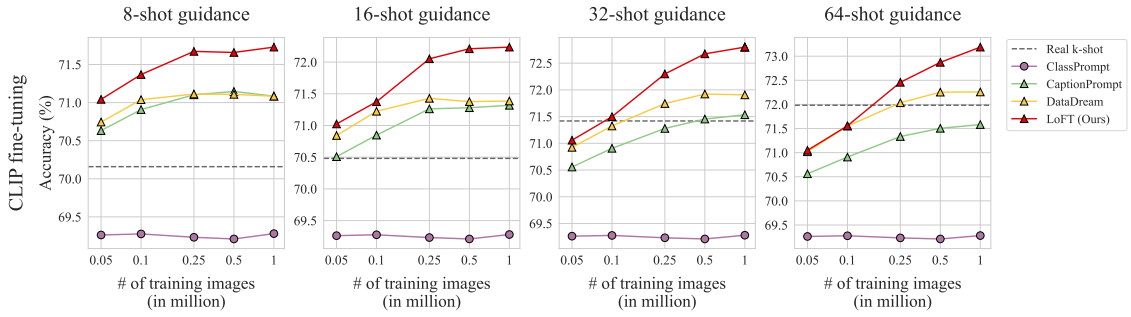


Figure 4.3: Classification accuracy on ImageNet when fine-tuning CLIP on synthetic data generated from different methods at different scales. We report few-shot guidance on 8, 16, 32, and 64 images per class and a baseline of training CLIP only on k-shot real data. LoFT consistently outperforms other methods and real k-shot result with small amount of synthetic data.

diffusion model objective while keeping the original parameters fixed:

$$\min_{\Delta W^{(i)}} \mathbb{E}_{\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_{\theta,\Delta W^{(i)}}(z_t, \tau(C(y_i)), t)\|_2^2\right], \tag{4.3}$$

where z_t corresponds to the noised x_i at step t and $C(y_i)$ is the template prompt `"a photo of a {`l_i`}"` with l_i being the class name of y_i. By learning LoRA weights for every few-shot image individually, LoFT overfits the diffusion model to a single sample, learning to reproduce all of its details even if such features were not in the training data of the original diffusion model. At inference time, every generated image will closely resemble the original real image, increasing fidelity, i.e., replicating fine-grained details.

Fusing LoRA weights. To increase the generation diversity from single-image LoRAs, we propose to interpolate their weights. We fuse two randomly selected LoRA weights corresponding to real images from the same class with

$$h_{\text{out}} = Wh_{\text{in}} + \lambda\Delta W^{(i)}h_{\text{in}} + (1-\lambda)\Delta W^{(j)}h_{\text{in}} \tag{4.4}$$

where $\lambda \in [0,1]$ and $\{(i,j)|y_i = y_j\}$.

This fusing strategy combines features from the different instances, effectively interpolating between real images in the weight space of the diffusion model, and improving the diversity of the generated images, as shown in the second phase in Figure 4.2. While $\lambda = 0$ or $\lambda = 1$ are reproducing the real data, choosing $\lambda = 0.5$ best interpolates images to produce new in-distribution samples of both high fidelity and diversity.

4.4 Experiments

We use Stable Diffusion [159] version 2.1 as the generative model for all synthetic dataset generation methods. For the ClassPrompt approach, we utilize the template prompt, `"a photo of a {`l`}"`. In the CaptionPrompt method, the prompt `"Caption the image:"`[1]

[1]Sourced from the original code base.

Method	Cal	DTD	Eur	Air	Pet	Car	SUN	Food	Flo	Avg
CLIP zero-shot	93.0	44.4	47.6	24.7	89.2	65.2	62.6	86.1	71.4	64.9
ClassPrompt	$93.9_{\pm 0.2}$	$53.7_{\pm 0.5}$	$46.2_{\pm 0.7}$	$26.5_{\pm 0.1}$	$92.5_{\pm 0.1}$	$74.1_{\pm 0.5}$	$67.0_{\pm 0.0}$	$85.1_{\pm 0.0}$	$71.9_{\pm 0.5}$	$67.9_{\pm 0.1}$
CaptionPrompt	$95.4_{\pm 0.3}$	$63.9_{\pm 1.0}$	$46.6_{\pm 1.7}$	$26.6_{\pm 0.1}$	$92.9_{\pm 0.1}$	$74.4_{\pm 0.1}$	$73.9_{\pm 0.2}$	$85.4_{\pm 0.0}$	$72.3_{\pm 0.5}$	$70.2_{\pm 0.1}$
DataDream	$96.0_{\pm 0.4}$	$64.9_{\pm 0.2}$	$84.1_{\pm 3.4}$	$61.4_{\pm 0.9}$	$\mathbf{93.5}_{\pm 0.2}$	$\mathbf{90.5}_{\pm 0.4}$	$74.4_{\pm 0.2}$	$\mathbf{86.5}_{\pm 0.0}$	$\mathbf{98.0}_{\pm 0.2}$	$83.2_{\pm 0.4}$
LoFT (Ours)	$\mathbf{96.7}_{\pm 0.2}$	$\mathbf{70.5}_{\pm 0.1}$	$\mathbf{86.8}_{\pm 2.2}$	$\mathbf{66.1}_{\pm 1.5}$	$93.2_{\pm 0.1}$	$89.3_{\pm 0.5}$	$\mathbf{75.5}_{\pm 0.0}$	$86.0_{\pm 0.0}$	$\mathbf{98.0}_{\pm 0.2}$	$\mathbf{84.7}_{\pm 0.2}$

Table 4.1: Classification accuracy on 9 fine-grained benchmarks when fine-tuning CLIP on synthetic data with 16-shot guidance. 500 synthetic images are generated for each class. Datasets are Cal: Caltech 101, Eur: EuroSAT, Air: FGVC Aircraft, Flo: Flowers 102.

Method	Cal	DTD	Eur	Air	Pet	Car	SUN	Food	Flo	Avg
CLIP zero-shot [155]	93.0	44.4	47.6	24.7	89.2	65.2	62.6	86.1	71.4	64.9
CoOp [243]	$95.5_{\pm 0.1}$	$67.8_{\pm 2.2}$	$78.9_{\pm 0.4}$	$38.7_{\pm 0.7}$	$93.3_{\pm 0.3}$	$78.3_{\pm 0.7}$	$74.0_{\pm 0.2}$	$86.7_{\pm 0.6}$	$95.8_{\pm 0.1}$	$78.8_{\pm 0.6}$
TIP-Adapter [238]	$95.1_{\pm 0.1}$	$65.4_{\pm 1.2}$	$77.6_{\pm 1.0}$	$39.4_{\pm 0.3}$	$91.8_{\pm 0.3}$	$75.6_{\pm 0.5}$	$72.1_{\pm 0.2}$	$86.5_{\pm 0.1}$	$94.6_{\pm 0.1}$	$77.6_{\pm 0.2}$
TIP-Adapter-f [238]	$95.8_{\pm 0.1}$	$72.2_{\pm 0.3}$	$89.0_{\pm 0.4}$	$44.9_{\pm 0.4}$	$93.0_{\pm 0.2}$	$83.3_{\pm 0.5}$	$76.3_{\pm 0.2}$	$\mathbf{87.3}_{\pm 0.0}$	$96.8_{\pm 0.2}$	$82.1_{\pm 0.0}$
AMU-Tuning [194]	$97.1_{\pm 0.3}$	$70.0_{\pm 1.0}$	$90.4_{\pm 0.4}$	$47.7_{\pm 1.6}$	$92.8_{\pm 0.1}$	$78.5_{\pm 0.1}$	$72.6_{\pm 0.2}$	$85.7_{\pm 0.2}$	$95.4_{\pm 0.2}$	$81.1_{\pm 0.3}$
LoFT (Ours)	$\mathbf{97.3}_{\pm 0.1}$	$\mathbf{73.8}_{\pm 0.5}$	$\mathbf{93.1}_{\pm 0.9}$	$\mathbf{71.8}_{\pm 1.6}$	$\mathbf{94.3}_{\pm 0.4}$	$\mathbf{90.7}_{\pm 0.3}$	$\mathbf{77.3}_{\pm 0.0}$	$87.2_{\pm 0.1}$	$\mathbf{99.2}_{\pm 0.0}$	$\mathbf{87.2}_{\pm 0.3}$

Table 4.2: Comparison between the state-of-the-art few-shot learning methods on 9 fine-grained benchmarks. CLIP ViT-B/16 is used as a base model with a 16-shot setting. Baseline methods use real data, and LoFT use real data as well as synthetic data for the training set. Datasets are Cal: Caltech 101, Eur: EuroSAT, Air: FGVC Aircraft, Flo: Flowers 102.

is used to caption each input image with PaliGemma [14]. Once a list of captions is generated, each caption is appended to the template prompt, forming prompts such as "`a photo of a {`l`}, {caption}.`" These prompts are then used as conditional input to the diffusion model to create synthetic images. For DataDream, We adopt the hyperparameter configuration [90] except that we exclude LoRA on text encoders for training since we found this to perform better. A guidance scale of 2.0 is used for all methods when generating synthetic images.

For our LoFT method, we employ AdamW [118] as the optimizer, a learning rate of 1e-3 with a cosine annealing scheduler, and a LoRA rank $r = 2$ for all trained LoRA adapters, which proved sufficient for adapting to single images. When generating images, LoFT fuses LoRA weights by randomly selecting two LoRA adaptations and fusing them with equal weights $\lambda = 0.5$. Different fusion strategies of LoRA weights are studied in §4.4.5.

4.4.1 Synthetic training data on ImageNet

To evaluate different dataset generation methods, we train an image classification model on each synthetic dataset. The target classification task is ImageNet [42], which contains 1,000 classes. For each generation method, we produce 50, 100, 250, 500, and 1,000 images per class, corresponding to dataset sizes of 0.05M, 0.1M, 0.25M, 0.5M, and 1M. We use these datasets to fine-tuning a pre-trained CLIP [155] model and evaluate it on the validation set of ImageNet. Our goal is to investigate whether synthetic training data can provide

additional useful information to improve the performance of a pre-trained model that already has some knowledge of the downstream task. Following the work of [37] and [90], we use the pre-trained CLIP ViT-B/16 model [155] as the base model, and fine-tune a LoRA (rank 16) applied to both the vision encoder and text encoder with the synthetic training data for ImageNet. We also conducted training ResNet50 [68] from scratch, which is shown in Appendix B. For synthetic data generation methods leveraging few-shot real images, we conduct experiments in 8-, 16-, 32- and 64-shot settings to examine performance under different guidance levels. The CLIP fine-tuning results are shown in Figure 4.3.

ClassPrompt does not scale. ClassPrompt dataset generation (purple line) improves over the baseline CLIP performance of 66.6%, and fluctuates between 69% and 70% as the dataset size increases. It indicates that the ClassPrompt method fails to improve as more images are generated. Consequently, synthetic images generated by the ClassPrompt method provide minimal additional information to the pre-trained CLIP model and showing no ability to scale.

Few-shot guided methods outperform ClassPrompt. Across all dataset sizes and k-shot settings, all few-shot guided methods ($\triangle$ markers in Figure 4.3) outperform ClassPrompt ($\circ$ markers). This is because few-shot guided methods generate higher-quality images with better diversity (for CaptionPrompt) or higher fidelity (for DataDream and LoFT). We provide a detailed analysis in §4.4.3 and a qualitative examples in §4.4.4. This indicates that the inclusion of real-image guidance in the synthetic data generation process significantly improves the quality of the synthetic training dataset for training the downstream model.

Few-shot guided methods outperform real k-shot. The dashed line in each plot represents the performance of a model trained solely on k real images per class. We observe that even with a small number of synthetic images, models trained on a few-shot guided synthetic dataset can easily outperform models trained with real k-shot data. For example, the accuracy of the 16-shot real data is 70.48%. By generating 50 synthetic images per class (resulting in dataset size 0.05M), the accuracy of the model trained on LoFT dataset can reach 71.02%.

LoFT effectively scales across different k-shot settings. Unlike ClassPrompt, few-shot guided methods show consistent improvement in performance as the dataset size increases, with LoFT achieving the best performance across all k-shot settings. For instance, in the 16-shot setting, LoFT shows a 1.22% performance improvement, increasing from 71.02% to 72.24% as the dataset size grows from 0.05M to 1M training images, while CaptionPrompt shows a 0.81% improvement (70.51% $\rightarrow$ 71.32%). The difference becomes bigger in the 64-shot setting where LoFT shows a 2.14% improvement (71.05% $\rightarrow$ 73.19%) while CaptionPrompt shows a 1.02% improvement (70.56% $\rightarrow$ 71.58%).

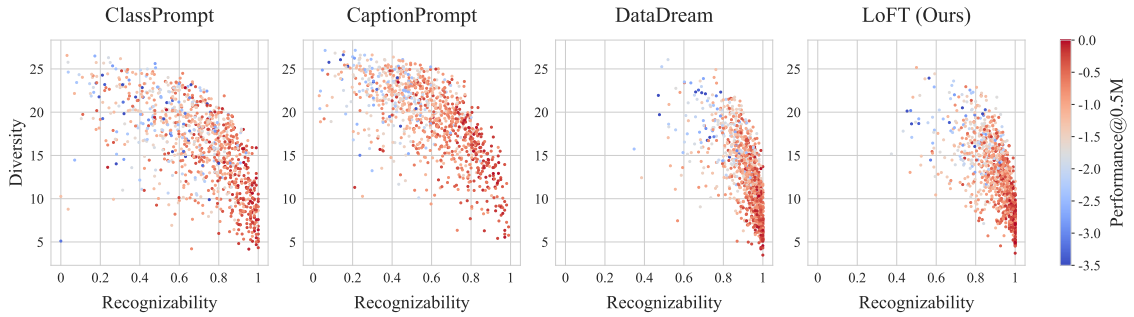


Figure 4.4: Per-class analysis on synthetic datasets generated from different methods. The color indicates a log-likelihood of the ImageNet validation dataset when CLIP is fine-tuned on the 0.5M-sized synthetic dataset in the 16-shot setting.

4.4.2 Synthetic training data on fine-grained datasets

4.4.2.1 Comparison of synthetic data generation methods

To further examine the effectiveness of our LoFT method, we evaluate on 9 fine-grained benchmarks: Caltech101 [105], DTD [36], EuroSAT [72], FGVC Aircraft [121], Oxford Pets [145], Stanford Cars [100], SUN397 [218], Food101 [17], and Flowers102 [139]. For each dataset, we generate 500 synthetic images for each class and fine-tune the CLIP model. For few-shot methods, we use 16-shot.

The results are shown in Table 4.1. We observe that our LoFT method outperforms other methods on 6 out of 9 benchmarks, achieving the best average accuracy of 84.7%. For instance, LoFT performs significantly better than DataDream on the DTD dataset (70.5% vs. 64.9%) which consists of texture images. This may attribute from LoRA by DataDream having challenges in learning texture patterns with a batch of images, whereas optimizing a single image using LoFT leads to better convergence, thus generating more in-distribution images in the generation phase. Additionally, ClassPrompt and CaptionPrompt underperform on the Aircraft and Cars benchmarks, due to the limitations of diffusion models in distinguishing fine-grained classes based on their class names. We further study when scaling the number of synthetic images in Appendix C. It shows that for LoFT, the scaling curve meets a plateau at 500 images per class on DTD, while it keeps increasing over 5000 images per class on the Aircraft dataset.

4.4.2.2 Comparison with Few-shot learning methods

We compare our method with state-of-the-art methods in the few-shot learning literature. CoOp [243] optimizes the learnable token of textual input, TIP-Adapter [238] designs a cache model from the few-shot training set, and AMU-Tuning [194] balances the CLIP logit with MOCOv3 [29]. We reproduce the results using the official source code of each. While the few-shot learning methods optimize a pre-trained model using real data only, with LoFT method, we use both the few-shot real images as well as 500 synthetic images per class for the training set. CLIP ViT-B/16 is used as a base model, and we conduct

experiments in the 16-shot setting.

The results are presented in Table 4.2. Our LoFT method outperforms baseline methods on 8 out of 9 benchmarks, achieving the best average accuracy of 87.2% vs. 82.1% of the next best method TIP-Adapter-f. We observe that there are significant performance gaps between LoFT and baseline methods on the Aircraft and Cars datasets, which consist of fine-grained classes. This indicates the advantages of using the synthetic data of LoFT in addition to the few-shot real images. Qualitative examples on these datasets can be found in Appendix D.

4.4.3 Per-class analysis on ImageNet

To identify how different factors in the synthetic dataset impact performance, following [55], we evaluate two metrics: recognizability and diversity. For our analysis, we randomly sample 50 images per ImageNet class for each synthetic dataset generation method.

- Recognizability: To evaluate the fidelity of the generated images, we use a pre-trained ImageNet ViT-B/16 classifier (accuracy of 86.2%) to classify the generated images. The F1 score for each class serves as the metric.
- Diversity: For each class, we extract features from the same pre-trained ImageNet ViT-B/16 classifier and compute their standard deviation as a measure of diversity in the generated images.

Figure 4.4 presents the scatter plots for each method where each point summarizes one class. The color of each point indicates the log-likelihood of the corresponding class in the validation set of ImageNet, as predicted by the CLIP model fine-tuned on 0.5M synthetic images in the 16-shot setting.

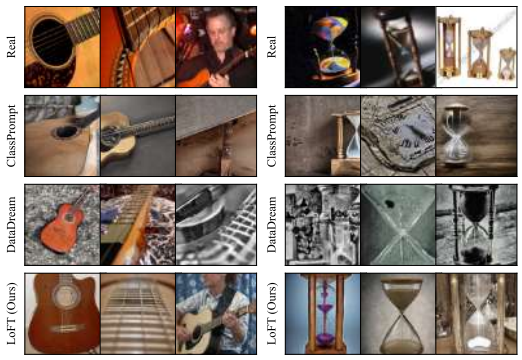
Recognizability and diversity are inversely correlated. Across all methods, there is an inverse correlation between recognizability and diversity: as recognizability increases, diversity tends to decrease, and vice versa.

Few-shot guided methods exhibit higher recognizability, while other methods have higher diversity. DataDream and LoFT show higher recognizability compared to ClassPrompt and CaptionPrompt. This is because these few-shot guided methods are specifically trained to generate images similar to the real images. This alignment improves the realism and quality of the generated images, leading to higher recognizability. While LoFT incorporates multiple LoRA weights to increase diversity, it still shows less diversity than ClassPrompt and CaptionPrompt.

Distinct strengths of each method. CaptionPrompt obtains a good performance on classes with high diversity (i.e., in the range of 20-25). On the other hand, DataDream and LoFT demonstrate better performance with high recognizability (i.e., when it is greater than 0.8). This suggests that each method has its own strengths. It remains an open question for further exploration of methods that combine these strengths, achieving high diversity and recognizability.

4.4.4 Qualitative comparison

We present qualitative results in Figure 4.5 to gain insights into the diversity and quality of images from different methods. For the acoustic guitar class (Figure 4.5a), real images have high variety, including differences in zooming, color palettes, and the presence of humans. In contrast, ClassPrompt images lack this diversity, displaying limited color variation and similar representations of the guitar. DataDream demonstrates a better level of diversity, generating images with various colors and textures. However, some DataDream images exhibit artifacts, which can detract from their overall quality. In contrast, our LoFT method successfully balances diversity and image quality. This is reflected in the quantitative metrics, where LoFT achieves the highest scores for recognizability (0.94) and diversity (14.68), outperforming ClassPrompt (0.88, and 9.54).



(a) Acoustic guitar (b) Hourglass

Figure 4.5: Qualitative examples for the classes Acoustic guitar and Hourglass from ImageNet. Our LoFT method generates diverse images, such as variations in zoom level, for acoustic guitar, and preserves an object of interest better for hourglass.

In the hourglass category in Figure 4.5b, ClassPrompt sometimes produces image related to "hour" but not "hourglass", as seen in the second column. Some of the images by DataDream show barely recognizable shapes of an hourglass. In contrast, images from LoFT closely resemble the object of interest. This observation is consistent with the quantitative metrics where LoFT achieves a recognizability score of 1.0 while ClassPrompt and DataDream score 0.89 and 0.95, respectively.

4.4.5 Ablation study of LoFT

Our ablation study investigates how different methods of fusing representations impact the synthetic training dataset. We conduct two studies: one exploring the effect of various fusion techniques, and another examining the influence of the weight parameter λ in our LoRA-based fusion method.

4.4.5.1 Fusion on different representations

To explore the effectiveness of different fusion techniques on the resulting synthetic dataset, we compare three methods for fusing representations: 1) caption embedding fusion, which involves averaging the text embeddings of the PaliGemma captions from two images; 2) image embedding fusion, which directly embeds two images using an image encoder, and

then passes the average image representation to Stable-Diffusion-2.1-unclip[2], a model for image-to-image generation; and 3) Textual Inversion [60] fusion, which optimizes input tokens for each image and then fuses the learned token embeddings from two images.

We generate up to 500 images per class on ImageNet using each method and fine-tune CLIP on the resulting datasets. Our results in the 16-shot setting are presented in Table 4.3, which shows that our LoRA-based fusion method outperforms the other techniques both when generating fewer samples (50K imgs, +0.66%) and with increasing number of generations (500K imgs, +1.55%). This suggests that our method is more effective at capturing the underlying structure of the data and generating high-quality images.

Fusing representation	0.05M	0.1M	0.25M	0.5M
Caption embeddings	70.36	70.55	70.62	70.66
Image embeddings	69.88	69.95	69.99	70.02
Tokens from Textual Inversion	69.70	70.13	70.21	70.29
LoRA weights (= LoFT, ours)	**71.02**	**71.37**	**72.05**	**72.21**

Table 4.3: Comparison of methods on fusing different representations when fine-tuning CLIP. Experiments are done in the 16-shot setting on ImageNet.

4.4.5.2 λ variation for LoRA fusion

We also examine the impact of λ on the generated images and downstream classification performance. In §G we show examples of images generated with different values of λ illustrating that $\lambda = 0.5$ provides the best visual results, especially in terms of diversity.

In Table 4.4, we quantitatively demonstrate the importance of choosing an optimal value for λ. When λ is set to 0.5, our method achieves the best performance, with a significant increase in accuracy as the amount of training data increases (scaling from 29.03% to 45.41% with 0.05M to 0.5M samples). In contrast, setting λ to values closer to 0 or 1 results in lower performance, due to a lack of diversity in the generated images (only scaling from 22.42% to 30.85% with 0.05M to 0.5M samples for $\lambda = 1$).

Fusing LoRAs	0.05M	0.5M
$\lambda = 0.5$	29.03	**45.41**
$\lambda = 0.7$ (or 0.3)	25.60	39.18
$\lambda = 1$ (or 0)	22.42	30.85
$\lambda \sim \text{Beta}(2, 2)$	28.28	43.15
$\lambda \sim \text{Beta}(5, 5)$	28.56	44.91
$\lambda \sim \text{Beta}(10, 10)$	**29.40**	45.36
$[0.5, 0.25, 0.25]$	27.87	43.63
$[0.33, 0.33, 0.33]$	28.46	44.10
$[0.7, 0.15, 0.15]$	22.19	36.28

Table 4.4: Ablation study on LoRA fusion. We train ResNet50 from scratch in the 16-shot setting.

Second, we explore the impact of introducing randomness to the λ value, using a Beta distribution $\text{Beta}(\alpha, \alpha)$, where α controls the concentration of λ around the value of 0.5. Larger values of α lead to a distribution that is more concentrated around 0.5, while smaller values of α allow for a broader spread across the [0,1] interval. When α is small, the performance decreases, i.e., accuracy of 43.15% ($\alpha = 2$) vs. 45.36% ($\alpha = 10$) with 0.5M

[2]https://huggingface.co/stabilityai/stable-diffusion-2-1-unclip

data samples. This suggests that having a more concentrated distribution around $\lambda = 0.5$ is beneficial for performance.

Finally, we also evaluate the performance of fusing three LoRA weights during dataset generation. Our results, presented in the last three rows of Table 4.4, show that none of these methods outperform the two-LoRA fusion with $\lambda = 0.5$. In fact, using more than two LoRAs introduces artifacts into the generated images, which can deteriorate their recognizability.

4.5 Conclusion

In this paper, we introduced LoFT, **Lo**RA-**F**used **T**raining-data Generation with Few-shot Guidance. LoFT fine-tunes LoRA weights on individual real images, ensuring high fidelity when generating synthetic images, and then fuses them to achieve diversity. Our experiments demonstrate that LoFT consistently outperforms other methods when fine-tuning a pre-trained CLIP model. This is because the synthetic images generated by LoFT complement the prior knowledge contained in CLIP by accurately capturing and fusing the features of each class. Additionally, we showed that LoFT performs better as the number of few-shot samples increases when training from scratch. Our analysis of the synthetic datasets showed that LoFT achieves a balance of high fidelity with reasonable diversity, while methods like ClassPrompt and CaptionPrompt focus more on generating diverse images at the cost of fidelity.

5

Improving Intervention Efficacy via Concept Realignment in Concept Bottleneck Models

Concept Bottleneck Models (CBMs) ground image classification on human-understandable concepts to allow for interpretable model decisions. Crucially, the CBM design inherently allows for human interventions, in which expert users are given the ability to modify potentially misaligned concept choices to influence the decision behavior of the model in an interpretable fashion. However, existing approaches often require numerous human interventions per image to achieve strong performances, posing practical challenges in scenarios where obtaining human feedback is expensive. In this paper, we find that this is noticeably driven by an independent treatment of concepts during intervention, wherein a change of one concept does not influence the use of other ones in the model's final decision. To address this issue, we introduce a trainable concept intervention realignment module, which leverages concept relations to realign concept assignments post-intervention. Across standard, real-world benchmarks, we find that concept realignment can significantly improve intervention efficacy; significantly reducing the number of interventions needed to reach a target classification performance or concept prediction accuracy. In addition, it easily integrates into existing concept-based architectures without requiring changes to the models themselves. This reduced cost of human-model collaboration is crucial to enhance the feasibility of CBMs in resource-constrained environments. Our code is available at https://github.com/ExplainableML/concept_realignment.

5.1 Introduction

Despite tremendous progress of Deep Learning (DL) techniques in research and applications, their adoption to high-stakes scenarios has been limited [97, 232, 234]. This is in large part due to unpredictable biases and failure cases of deep models when transferring to unseen data or complex & ambiguous cases grounded in the numerous model parameters, architecture designs and training choices [18, 23, 35, 47, 51, 54, 62, 127, 163].

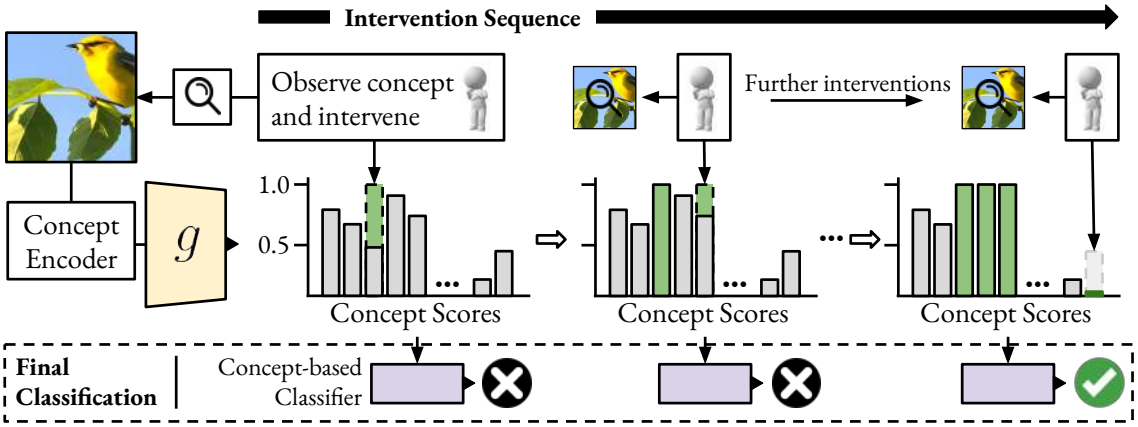


Figure 5.1: Concept-based classification models allow for human intervention, where a human expert can correct specifically assigned concepts. However, to achieve satisfactory performance, concept-based classification models often require a large number of interventions, where each additional intervention requires costly human interaction.

The black-box nature of typical DL models and their representation spaces [20, 23, 116, 161, 188] further exacerbate this problem, as it makes understanding and debugging the decision-making process of these models difficult. Consequently, it becomes hard for human practitioners to trustworthily operate these models in scenarios with significant legal [53, 206] or ethical [50, 150] constraints.

To foster trust, transparency in the decision-making process, and the ability to operate alongside expert feedback are required. In order to incorporate these desiderata into the design space of deep models, *Koh et al.* [97] introduced Concept Bottleneck Models (CBMs). These models break the decision process into the extraction of human-interpretable concepts (such as "white wings" and "orange beak" when classifying a seagull) from a given input, and a subsequent concept-grounded classifier operating on top of these concept predictions. While this allows users to peek into the model decision process - maybe even more importantly, it also uniquely allows for human-guided intervention and feedback integration at test time. This is done through concept interventions, wherein an expert user analyses predicted concepts and optionally replaces those they deem incorrect with ground-truth information (Fig. 5.1).

Such interventions can significantly raise the performance and reliability of these models [97, 184, 232, 234], while offering a natural interface for human-AI collaboration. However, human annotation is expensive, especially when resources and access to expert knowledge are limited. Ideally, such concept models should operate well with minimal human input. This becomes particularly prevalent as CBMs (as well as follow-up extensions such as Concept Embedding Models (CEMs, [232])) often require numerous interventions in order to significantly boost model performance [97, 232, 234], as the set of concepts these models operate on can often be rather extensive. For example, on the widely used CUB benchmark (bird classification, [207]), it takes 13 interventions per image on overage to raise the accuracy of a baseline CBM model from around 68% to 90% (Fig. 5.4).

In this work, we posit that a large part of this limited intervention efficacy can be traced back to the independent nature of how concept interventions are treated. This means that correcting for one concept (or a set of concepts) does not affect which other concepts are predicted for the same image. However, the occurrence of concepts in real life is often correlated, and informing the model about one concept should consequently influence the use of related ones. Not doing so means that we do not leverage human feedback to its full extent - intervening on one specific concept naturally gives additional context about the potential occurrence of other concepts, which should be taken into account in the final classification process. In particular, we study the extent of this crucial aspect when operating with concept-based models. Our study highlights how the use of a **simple *concept intervention realignment* module**, which learns from statistical concept relations, can effectively and automatically realign concept values after an intervention (or multiple) have been performed. Our experiments reveal how our concept intervention realignment can seamlessly integrate into and improve any existing concept-based approach (e.g. default CBMs [97], advanced CEMs [232] or recently introduced intervention-aware CEMs [234]), and can be deployed both jointly during the initial training of the concept model, and as a post-hoc trained realignment mechanism. Across three standard, real-world benchmarks (CUB [207], CelebA [115] and AwA2 [217]), we showcase consistent, in parts very significant improvements in intervention efficacy. Across both concept prediction accuracy as well as overall classification accuracy, performance increases more rapidly with interventions as compared to a baseline where concepts are not realigned (in parts reducing the number of interventions needed to reach a target performance by over 70%). Combined with its versatile usage and the minimal additional resource requirements, we believe our insights into concept intervention realignment to be of high practical relevance, helping to drive down the cost of human-model collaboration and facilitate the corresponding practical deployment of concept-based models.

5.2 Related Works

Concept Bottleneck Models (CBMs) have been extensively studied since their introduction by [97]. [232] proposed Concept Embedding Models as a generalization, utilizing embedding vectors for concepts rather than scalar probabilities, thus enhancing task performance while maintaining interpretability. Recent efforts have explored methods to enhance CBMs without requiring explicit concept supervision during training, leveraging pre-trained vision backbones and language guidance [142, 221, 230]. [4] introduced Self-explaining Neural Networks (SENNs) for unsupervised concept learning, while [174] proposed CBM-AUC combining SENNs with CBMs. Probabilistic CBMs [87] were proposed to model uncertainty in concepts and final predictions. [120, 126] addressed concept leakage, while techniques proposed by [67, 125] aimed to alleviate it. Our work complements these efforts by enabling CBMs to update predictions of all concepts after human intervention.

Interventions on CBMs. [97] showed that intervening on randomly selected concepts enhances classification performance in CBMs. [25] and [181] proposed uncertainty-based strategies for expert interventions. [184] extensively studied concept selection strategies, focusing on task performance and execution cost. [234] introduced interventions during training to enhance model receptiveness to test-time interventions. Our approach complements existing methods by updating predictions of all concepts following expert interventions, allowing integration with prior strategies. Concurrently, [220] proposed Energy-based CBMs to automate concept prediction updates. In comparison, our method benefits from higher simplicity, improved performance, and seamless integration with existing CBM approaches.

5.3 Methods

5.3.1 Background and Preliminaries

Concept Bottleneck Models. A Concept-Bottleneck Model (CBM) can be viewed as a composition of two models, $h = f(g(x)) : \mathcal{X} \rightarrow \mathcal{Y}$, with concept encoder $g : \mathcal{X} \rightarrow C$, and concept-based classification head $f : C \rightarrow \mathcal{Y}$. $\mathcal{X}, \mathcal{Y}$, where C denote input, class label, and concept sets, respectively. CBMs get their name from an inherently bifurcated optimization process: While the concept encoder $g(x)$ is trained to predict concepts $\hat{c} \in \mathbb{R}^k$ from the concept set with $\|C\| = k$ concepts given an image $x \in \mathbb{R}^d$, the classification head $f(\cdot)$ is optimized to predict final target labels $y \in \mathcal{Y} \in \mathbb{R}^M$ solely based on concept assignments produces by g. CBM training data is thus given as $\mathcal{D} := \{x^{(i)}, c^{(i)}, y^{(i)}\}_{i=1}^N$, where $x^{(i)}, c^{(i)}, y^{(i)}$ are the inputs, ground-truth concepts, and ground-truth labels, respectively. Following existing works [97, 232, 233, 234], the concept encoder g is trained using a (weighted) binary cross-entropy loss ($\mathcal{L}_{\text{concept}}(\hat{c}, c)$), while the classification head f utilizes a cross-entropy classification objective ($\mathcal{L}_{\text{task}}(\hat{y}, y) = \mathcal{L}_{\text{CE}}(\hat{y}, y)$).

Overall, there are three established schemes [97] for training CBMs: (1) *Independent training:* the concept encoder and classification head are trained entirely independently, with ground-truth concepts c provided as inputs to the classification head during training. (2) *Sequential training:* the concept encoder g is trained first, followed by the classification head f trained using the concepts predicted by g. (3) *Joint training:* both the concept encoder g and the classification head f are trained together using a combination of $\mathcal{L}_{\text{concept}}$ and $\mathcal{L}_{\text{task}}$, respectively. In all cases, this means that the classification head leverages only information on concept (co-)occurences to predict final class labels, making it easy to ground the final classification decision on interpretable concept assignments.

Concept Embedding Models. The flow of information in a CBM is bottlenecked by the set of user-defined concepts. This can potentially limit the processing capacity of the model, especially when the concepts do not contain all the information that is needed to

perform the downstream task. To overcome this issue, [232] proposed Concept Embedding Models (CEMs) as a generalization of CBMs wherein every concept i is represented by a pair of high-dimensional vectors, $\hat{\mathbf{c}}_i^+$ and $\hat{\mathbf{c}}_i^-$ (as opposed to scalar concepts in CBMs). These embeddings are generated by passing x through concept-specific networks ϕ_i^+ and ϕ_i^-, and represent the concept being present and absent, respectively.

The probability $\hat{p}_i$ of the concept i being in x is then simply computed by passing $\hat{\mathbf{c}}_i^+$ and $\hat{\mathbf{c}}_i^-$ to a scoring function s as $\hat{p}_i = s([\hat{\mathbf{c}}_i^+, \hat{\mathbf{c}}_i^-])$. Similarly, both embeddings can also be combined as $\hat{\mathbf{c}}_i = \hat{p}_i\hat{\mathbf{c}}_i^+ + (1-\hat{p}_i)\hat{\mathbf{c}}_i^-$ to parameterize a joint embedding for concept i. The final concept embedding which represents the full image x and is passed to the classification head is then given as $\hat{\mathbf{c}} := [\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2, ..., \hat{\mathbf{c}}_k]$. Notice the much higher dimensionality of the concept embedding, which concatenates k concept-specific embeddings (as opposed to producing just a single k-dimensional concept vector).

Concept Interventions. Both CBMs and CEMs allow users to intervene on concepts at test time. Concretely, starting from the concept predictions of the model, $\hat{c}$, the user sequentially intervenes on $T \leq k$ concepts. As a human expert has to both investigate concept predictions and compare against input data, interventions are difficult to parallelize, effectively equating concept intervention into a trajectory of T concept intervention steps [184] (see also Fig. 5.1 for intuition).

Let $\mathcal{S}_t$ represent the set of concepts that have been intervened on up to time $t \leq T$. The corresponding concept embedding at time t is then given as $\tilde{c}_t = \{c_{\mathcal{S}_t}, \hat{c}_{\setminus \mathcal{S}_t}\}$, where $c_{\mathcal{S}_t}$ denotes the ground truth values of the intervened concepts, and $\hat{c}_{\setminus \mathcal{S}_t}$ are the model's predictions of non-intervened concepts. Intervening on concepts in this way updates the final prediction of the model from $\hat{y}$ to $\tilde{y} = f(\tilde{c})$. In the case of a CEM, intervening on concept i to update its value from $\hat{p}_i$ to p_i changes its embedding from $\hat{\mathbf{c}}_i$ to $\tilde{\mathbf{c}}_i = p_i\hat{\mathbf{c}}_i^+ + (1-p_i)\hat{\mathbf{c}}_i^-$.

After each intervention t, we use a concept intervention policy $\pi(\tilde{c}_t)$ to decide which concept to intervene on next. While π can simply suggest random concepts for intervention, it is often much better to leverage heuristics that rank concepts in the order of importance (by some measure). A commonly deployed, effective intervention policy is UCP [103, 184], which utilizes the uncertainty of concepts. In particular, UCP selects concepts with the highest uncertainty, i.e. concept predictions closest to 0.5. More details about the intervention process can be found in Algorithm 1.

Intervention-aware CEMs. While test-time interventions typically improve performance, this is not always guaranteed. In fact, recent works have shown that concept interventions can in some cases even hurt the model's performance [184, 233]. [234] noted that this stems from the lack of training incentive for the model to perform well under intervention. To address this, they proposed Intervention-aware CEMs (IntCEMs), which introduce interventions during the training process to improve the model's receptiveness to interventions at test time, outperforming all existing methods in the intervention setting. In

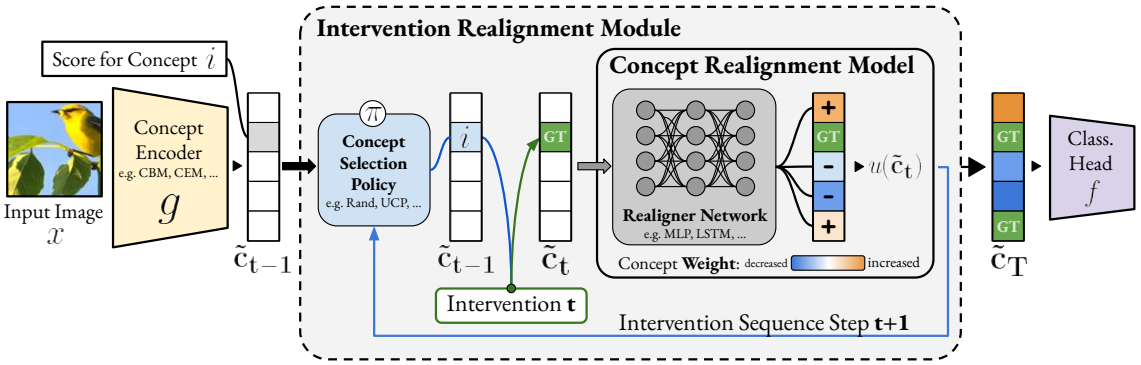


Figure 5.2: **Illustration of the concept intervention realignment module.** Given the concept encoding $g(x)$, we intervene on the concept i selected by a concept selection policy π. This concept is replaced with a ground-truth (GT) value ($\in \{0, 1\}$ depending on whether it is present in a given image or not) to obtain $\tilde{c}_t$ (representing intervention step $t \in \{1, ..., T\}$). This intervened concept representation is then passed into the concept realignment module (leveraging e.g. an MLP or LSTM reweighting mode), which outputs the realigned $u(\tilde{c}_t)$. To ensure that the ground-truth values provided by the user are not overwritten during realignment, $u(\tilde{c}_t)$ retains ground-truth corrections. The final concept vector is then based into a concept-based classifier f.

particular, they train a CEM to minimize the following objective:

$$\mathcal{L}_{\text{IntCEM}}(x, c, y, \mathcal{T}) = \mathcal{L}_{\text{pred}}(x, c, y, \hat{c}, \tilde{c}_t) + \lambda_{\text{conc}}\mathcal{L}_{\text{conc}}(\hat{c}, c) + \lambda_{\text{roll}}\mathcal{L}_{\text{roll}}(x, c, y, \mathcal{T}) \quad (5.1)$$

$$\mathcal{L}_{\text{pred}}(x, c, y, \tilde{c}_0, \kappa_t) = \frac{\text{CE}(f(\hat{c}, y) + \gamma^T \text{CE}(f(\tilde{c}_t), y)}{1 + \gamma^T}$$

$\mathcal{L}_{\text{pred}}$ is the prediction loss for y, $\mathcal{L}_{\text{conc}}$ the concept prediction loss, $\mathcal{L}_{\text{roll}}$ the rollout loss incentivizing the model to predict the most informative concept for intervention. λ_{conc} and λ_{roll} are user-defined weights corresponding to $\mathcal{L}_{\text{conc}}$ and $\mathcal{L}_{\text{pred}}$ respectively, while $\mathcal{T}$ denotes the intervention trajectory. $\mathcal{L}_{\text{pred}}$ penalizes the model for incorrect predictions both before and after the intervention, and $\gamma \geq 1$ is a scaling term that prioritizes correct predictions after intervention.

5.3.2 Concept Intervention Realignment

Previous works incrementally improve on predecessor methods by better parameterizing concept representations or introducing an intervention-aware training objective. However, all these works still treat concept interventions independently. This means that an intervention on one specific concept has no effect on the assignment of other concepts. This disregards relationships between concepts, which in practice do not occur independently (e.g., "white wing" and "white belly" are more likely to co-occur). As a result, the existing intervention process does not utilize human feedback optimally, as information about the verified existence of one concept should naturally guide the prediction of other concepts. While this aspect is naturally important to ensure that an accurate concept representation

is passed to the label classifier, it is also crucial when utilizing concept-selection criteria such as UCP because intervening on one concept should consequently reduce the chances of intervening on other closely related, likely co-occurring concepts, while also raising the probability that uncertain and unrelated concepts get intervened on.

Intervention Realignment Module. To address this, we propose a *concept intervention realignment module* (**CIRM**), which consists of two interdependent components: (a) a *concept realignment model* (**CRM**), $u : C \rightarrow C$. After a user intervenes on a subset of concepts $\mathcal{S}$, the remaining concepts ($\setminus\mathcal{S}$) are updated by a realigner network; and (b) an *intervention policy* π. The concepts predicted by the realignment model are fed to the policy to suggest which concept to intervene on next. Both components are interdependent, and together form the overall concept intervention realignment module, as also visualized in Figure 5.2. The training of the full CIRM comprising both selection policy and concept realignment model aims to simulate the complete intervention process. It thus starts from the concept predictions of the base model, $\hat{c}$, where we sequentially intervene on concepts for $T \leq k$ time steps by following a policy of choice, π (in our case UCP by default, which we experimentally find to outperform random intervention significantly; See Supp. §B).

As in §5.3.1, let $\mathcal{S}_t$ denote the set of intervened concepts and $\tilde{c}_t = \{c_{\mathcal{S}_t}, \hat{c}_{\setminus\mathcal{S}_t}\}$ denote the concepts at time t, respectively. At every intervention time step, we feed $\tilde{c}_t$ to the realignment model to obtain updated concept predictions as $\kappa_t = u(\tilde{c}_t)$, which in turn are utilized by $\pi(\kappa_t)$ to produce intervention recommendations for $t+1$. Finally, we train u with the ground-truth labels as targets using the loss $\mathcal{L}(u) = (\sum_{t=0}^{T} \text{CE}(u(\tilde{c}_t), c))/T$.

Using this simple objective, the concept realignment model u learns to take concept representations and leverage intervened concepts $\mathcal{S}_t$ to predict an updated concept distribution, i.e., $p(i; \hat{c}, \mathcal{S}_t)$. Note that this training objective utilizes standard CBM training information (i.e., concept annotations, [97, 184, 232, 233, 234]); so no additional information beyond the standard CBM pipeline is required.

The overall training pipeline can still follow the standard CBM training paradigms (see previous section), with the intervention realignment module being trained independently on top of a pre-trained frozen CBM/CEM as a posthoc realignment method, or jointly with the CBM/CEM to introduce an explicit realignment objective during training. For posthoc realignment, we first train the backbone f and the classification head g. Subsequently, we freeze those components and train the realignment model u.

Realignment Models. As shown in Fig. 5.2, we parameterize our concept realignment model with a neural network v. To ensure that u does not overwrite the ground-truth concepts provided by the user, we also keep track of the already intervened concepts $\mathcal{S}_t$. Using this information, we replace the output of the realigned concept embedding with the user-provided values for concepts in $\mathcal{S}_t$. Hence, the final output of u for the i^{th} concept

is given as

$$u(\tilde{c}_t, \mathcal{S}_t)^{(i)} = \begin{cases} v(\tilde{c}_t)^{(i)} & \text{if } i \notin \mathcal{S}_t \\ \tilde{c}_t^{(i)} & \text{if } i \in \mathcal{S}_t. \end{cases}$$

Depending on the assumptions made on the realignment process, v is either a simple MLP or a recurrent model (such as an LSTM [78]). The former parametrizes our default concept intervention realignment model, which only passes the set of intervened and un-intervened concepts at intervention step t to the concept realignment model consisting of a simple MLP. The set of concepts fed into the MLP may either be the original concept embedding $\tilde{c}_0$, where all intervened concepts up to and including step t have been replaced with ground-truth values, or the previously realigned κ_{t-1} with similarly updated intervened concepts (c.f. Fig. 5.2, "GT"). Note that in either case, κ_{t-1} informs the selection process of the subsequent concept to intervene on. After all interventions, the final concept embedding fed into the classifier is always $u(\tilde{c}_T)$. Practically, we found using $\tilde{c}_t$ to work slightly better than κ_{t-1}. Both cases above however only pass the final set of concepts at time t to the realignment model. Given the sequential nature of interventions, however, it may also be beneficial to account for the entire intervention history to inform future concept realignment. As a result, we also introduce a recurrent realignment variant, u_{rec}, which employs an LSTM model to retain the entire history of interventions until time t. An algorithmic summary is provided in supplementary §A.

End-to-End Realignment. In order to jointly train the CIR module and the base model $f \circ g$, we will perform interventions while also training the base model. This naturally combines with the IntCEM framework [234], which incorporates train-time interventions, and as such is our default choice for joint model and realignment module training.

Concretely, we modify IntCEMs such that after t interventions, concepts $\tilde{c}_t$ are corrected post-intervention to obtain $\kappa_t = u(\tilde{c}_t)$, which is then fed to the classifier f. The new training objective, then, is:

$$\mathcal{L}_{\text{IntCEM-ReA}}(x, c, y, \mathcal{T}) = \mathcal{L}_{\text{pred}}(x, c, y, \tilde{c}_0, \kappa_t) + \lambda_{\text{conc}} \mathcal{L}_{\text{conc-ReA}}(\hat{c}, c, \kappa_0, \kappa_t) + \lambda_{\text{roll}} \mathcal{L}_{\text{roll}} \quad (5.2)$$

$$\mathcal{L}_{\text{conc-ReA}}(\hat{c}, c, \kappa_0, \kappa_t) = \frac{1}{2} \left(\mathcal{L}_{\text{conc}}(\hat{c}, c) + \frac{\text{CE}(\kappa_0, c) + \gamma^T \text{CE}(\kappa_T, c)}{1 + \gamma^T} \right) \quad (5.3)$$

where $\mathcal{L}_{\text{conc-ReA}}$ is the modified concept prediction loss which trains both the backbone g of the base model (first term) as well as the CRM (second term). We use the same γ as in $\mathcal{L}_{\text{pred}}$ to prioritize correct predictions by the CRM after intervention, and the same λ_{conc} and λ_{roll} as in Eq. 5.1.

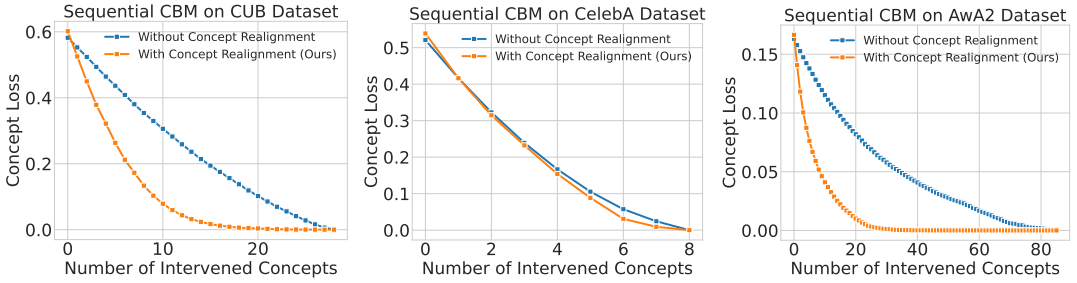


Figure 5.3: Concept prediction loss vs. the number of intervened concepts with and without concept realignment. Concept realignment consistently improves concept predictions.

5.4 Experiments

5.4.1 Preliminaries

Datasets. We perform experiments on three datasets: (1) **Caltech-UCSD Birds-200-2011 (CUB)** [207] containing $n = 11,788$ bird images over 200 classes. Following the original CBM paper [97], we use 112 concepts grouped into 28 concept groups with the same splits. (2) **Large-scale CelebFaces Attributes (CelebA)** [115] contains over 200,000 celebrity images annotated with 40 attribute labels, including noisy characteristics such as gender and age. Following [232, 234], we use only the most balanced 8 concepts in our experiments, resulting in $2^8 = 256$ classes. (3) **Animals with Attributes 2 (AwA2)** [217] is a collection of $n = 37,322$ animal images over 50 classes annotated with 85 attributes such as species, color, and behavior.

Implementation Details. We perform experiments on CEMs, IntCEMs, and three types of CBMs (sequential, independent, and joint). For all models and datasets, we follow the hyperparameters used in [234]. During CIRM training, we sequentially intervene on concepts $T = k$ times. By default, we use UCP both during training and inference, and if not stated otherwise, use a multi-layered perceptron (MLP) for concept realignment. We use the predictions of the base CBM ($\tilde{c}_t$) as its input for un-intervened concept representations. We perform a small, standard hyperparameter using Optuna [2] with 50 trials to search over the number of hidden layers $\in \{1, 2, 3\}$ and units $\in \{k, 2k, k/2\}$, the learning rate $\in [10^{-5}, 10^{-1}]$ and weight decay $\in [10^{-6}, 5 \times 10^{-5}]$, and use the same batch size as used to train the base model. We employ early stopping and learning rate decay on the validation loss. For joint training, we instantiate the realigner MLP 2 hidden layers containing k neurons each. Experiments are conducted using PyTorch [146].

5.4.2 Concept Realignment Improves Intervention Efficacy

To probe the efficacy of our concept intervention realignment module, we evaluate both the change in concept prediction loss as well as overall classification accuracy as a function of intervened concept counts. These are visualized in Fig. 5.3 and Fig. 5.4 for sequentially

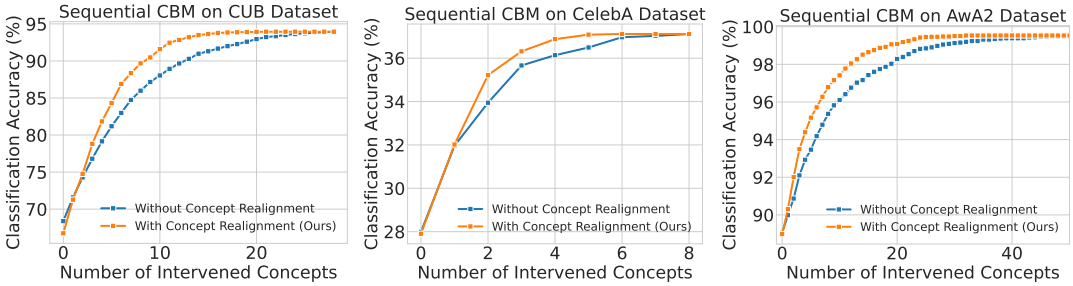


Figure 5.4: Classification accuracy vs. the number of intervened concepts with and without concept realignment. Realignment consistently improves classification accuracy.

Table 5.1: Area Under Curve (AUC) of Concept Prediction Loss and Classification Accuracy with/without CIRM. We use the same backbone for sequential and independent CBMs. CIRM improves performance across all models and datasets. Intervention curves share long saturation plateaus for high intervention counts. Accuracy AUC scores are thus saturated, and best combined with performance graphs in Figs. 5.3, 5.4.

Base Model	**Realigned**	**Concept Loss AUC ↓**			**Accuracy AUC ↑**		
		CUB	CelebA	AwA2	CUB	CelebA	AwA2
Sequential CBM	×	6.71	1.59	4.26	2460.8	280.7	8364.0
	✓	**3.15**	**1.52**	**1.13**	**2510.9**	**284.3**	**8397.6**
Independent CBM	×	6.71	1.59	4.26	2653.4	280.2	8403.4
	✓	**3.15**	**1.52**	**1.13**	**2678.3**	**282.1**	**8437.0**
Joint CBM	×	5.93	3.06	4.77	2580.3	273.1	8276.4
	✓	**3.67**	**1.76**	**1.48**	**2609.0**	**273.9**	**8327.4**
CEM	×	5.99	1.61	4.90	2521.4	396.3	8429.3
	✓	**3.20**	**1.46**	**1.69**	**2558.4**	**400.1**	**8433.9**

trained CBMs (see §5.3.1), respectively, for all benchmark test sets - CUB, CelebA and AwA2. Note that for AwA2, we only show the first 50 interventions for visual clarity, as performance beyond that heavily plateaus since sufficient concepts have been intervened on to perfectly solve the test data. Table 5.1 numerically summarizes these results via AUC scores and provides additional scores for independently trained CBMs, jointly trained CBMs as well as Concept Embedding Models (CEMs). Runs in Tab. 5.1 and Figs. 5.3, 5.4 all utilize the stronger UCP concept selection policy as opposed to the weaker random selection policy (§B) to measure intervention efficacy at the highest level, and train the concept realignment module on top of already trained concept models.

Improved concept attribution through intervention. Across all datasets, we can observe a consistent, in parts vast reduction in concept prediction loss, which measures the correct assignment of concepts for each input (using the concept loss described in §5.3.1). For example on CUB, a *tenfold* reduction of the original unintervened concept loss (~ 0.6 to ~ 0.06) can be achieved with half the number of interventions (11 with concept realignment, 23 without). This effect becomes even more prevalent on AwA2, where a tenfold reduction ($\sim 0.17 \rightarrow \sim 0.017$) is achieved after around 16 interventions with realignment versus more than 60 without; marking a more than 70% reduction in intervention efforts. This

is also reflected in Tab. 5.1, where concept loss AUC drops by in parts more than half for CUB and from 4.26 to 1.13 on AwA2. We find this significant improvement in concept attribution persists across all CBMs and CEMs, as well as random seed initializations (see Supp. Tab. D.1, D.2 and D.3)

We do find that for CelebA with a much more restrictive concept bottleneck than e.g. CUB and AwA2, due to significantly fewer (note that in CUB concepts are already grouped, see §5.4.1) and noisier concepts, that the overall gain in concept accuracy is smaller. This is also reflected in the notably weaker performance of the base CBM (c.f. Fig. 5.4, middle - less than 38% accuracy when intervening on *all* concepts), which strongly points towards overall insufficient concept information provided in the CelebA training data. Overall, however, we find very clear evidence that the concept intervention realignment module allows practitioners to leverage human intervention feedback to a much larger extent to attribute the correct concepts to respective inputs. This means that the subsequent classifier will operate on a much more accurate set of concepts, thereby improving the overall interpretability of the final classification decision.

Improved overall classification through intervention. On top of that, we also find that the significant gain in intervention efficacy on a concept attribution level also translates to subsequent gains in intervention efficacy for the overall classification performance (Fig. 5.4). For example on CUB, the final classification accuracy after intervening on all concepts is 93.9%, which is achieved already after $\sim$ 16 intervention steps. A comparable performance without concept intervention realignment requires nearly complete, $\sim$ 24 intervention steps, marking a 50% increase. The same can be seen on CelebA and AwA2 as well, where the upper-bound performance can be achieved with much fewer interventions (particularly without the need to intervene on *all* concepts). Even intermediate performance targets are achieved much earlier; a classification accuracy target of e.g. 98% on AwA2 requires only 12 concept interventions with realignment, while the non-aligned baseline needs 19 interventions on average. We find these results to be also reflected numerically in Tab 5.1, where accuracy AUC increases from e.g. 2460.8 to 2510.9 on CUB. We do point towards high numerical saturation given the larger performance plateaus at higher intervention counts, and high starting accuracies (e.g. $\sim$ 90% on AwA2). Numerical results are thus best considered alongside the intervention trajectories in Figs. 5.3 and 5.4.

Together, our experiments provide strong evidence that concept intervention realignment is crucial to best leverage human feedback in concept-based decision systems; allowing to significantly reduce intervention budgets by in parts over 70% to achieve a desired target performance. These gains can also be achieved *after* concept models have been trained, allowing for versatile applicability.

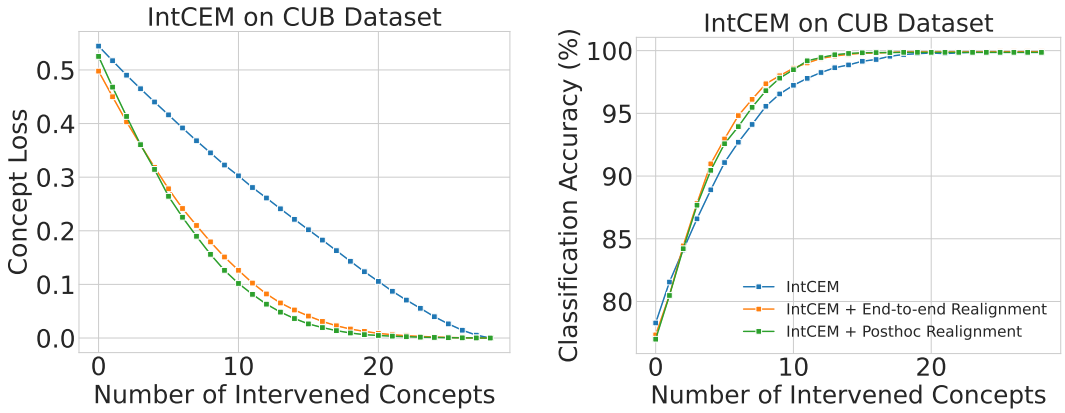


Figure 5.5: Concept Intervention Realignment in intervention-aware CEMs. (a) Concept prediction loss and (b) classification accuracy with jointly and post-hoc trained CIRMs. In both cases, significant benefits can be seen, especially for correct concept attribution after intervention - both for jointly and posthoc trained realignment modules.

5.4.3 Intervention Realignment for Intervention-aware CEMs

In this section, we investigate training the CIRM during the training process of an already intervention-regularized concept model; namely the recently proposed, state-of-the-art intervention-aware CEM [234] (see also §5.3.1). Following the objective described in Eq. 5.2, we operate and train the concept intervention module in conjunction with the intervention objective proposed by [234].

Our results are shown in Fig. 5.5. First, we find that explicit concept intervention realignment can significantly improve correct concept attribution, even in intervention-aware training setups (c.f. Fig. 5.5a). While not as significant as improvements over standard CBM models, for specific target concept prediction losses (such as a *fivefold* reduction from 0.5 to 0.1), half the number of intervention steps are needed (11 versus 20). The improved concept attribution is also reflected in higher intervention accuracies as seen in Fig. 5.5b, albeit the overall (still notable!) improvement is less reflective of the significant gains on a concept level (additional results can be found in Supp. §C). Overall, however, our experiments highlight that even when applied to state-of-the-art approaches that specifically simulate the intervention process during training, improved intervention efficacy can be found. Importantly, the consistently significant improvements on a concept attribution level mean that classification decisions are much better grounded on correct concept attributions, which is crucial for interpretability [97, 232] of classification results. Finally, we find that concept intervention realignment can be applied both as a regularization mechanism during training, as well as adapted entirely posthoc, while still offering consistent benefits. This supports the high versatility of CIRMs as a general-purpose tool to increase intervention efficacy.

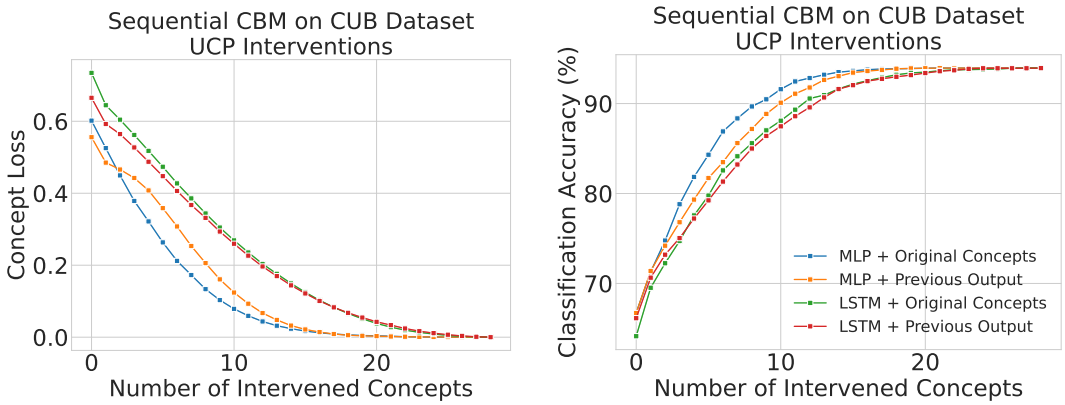


Figure 5.6: (a) Concept prediction loss and (b) classification accuracy for various realigner architectures alongside UCP policy. Using an MLP with concept predictions of the base model works better than compounding refinements and accounting for intervention trajectories using LSTMs.

5.4.4 Realignment Module Ablations

Realignment Model Architectures. In this section, we study the effect of various design choices for the realignment module along two dimensions: **(1) Recurrent vs. Feedforward Networks:** Since we intervene on concepts sequentially, it is possible that the realignment module can benefit from the overall order and history of interventions to make more accurate concept predictions. To do this, we instantiate the concept realignment network using an LSTM [78]. We compare this against our default MLP. **(2) Previous Output vs. Original Concepts:** By default, the realignment module takes as input a combination of ground-truth concepts provided by the user and values predicted by the base model at $t = 0$ for the concepts that have not been intervened on (see also §5.3.2). Due to the sequential nature of interventions, one may also directly feed the output of the realignment module at time $t - 1$ as input to it at time t in order to compound the refinements over multiple time steps. Combining both axes results in four recombinations, which we compare in Fig. 5.6. As can be seen, there is limited gain when accounting for the complete intervention history using an LSTM realigner network. Similarly, we find that applying the MLP primarily for concept selection alongside UCP and as final input to the classification head works better than compounding refinements over intervention steps.

Intervention Policy Transfer. In this section, we study the importance of aligning intervention policies used during training with those deployed at test time. In particular, we operate on the base setup, which deploys the CBM and the concept intervention realignment module using only the much weaker random intervention policy at test time. However, we change the policy used to train the concept intervention realignment module. Our results are visualized in Fig. 5.7. As can be seen, while a realignment module trained with UCP can still be effective when deployed with a random intervention policy, it is notably outperformed by the weaker random policy at test-time when the

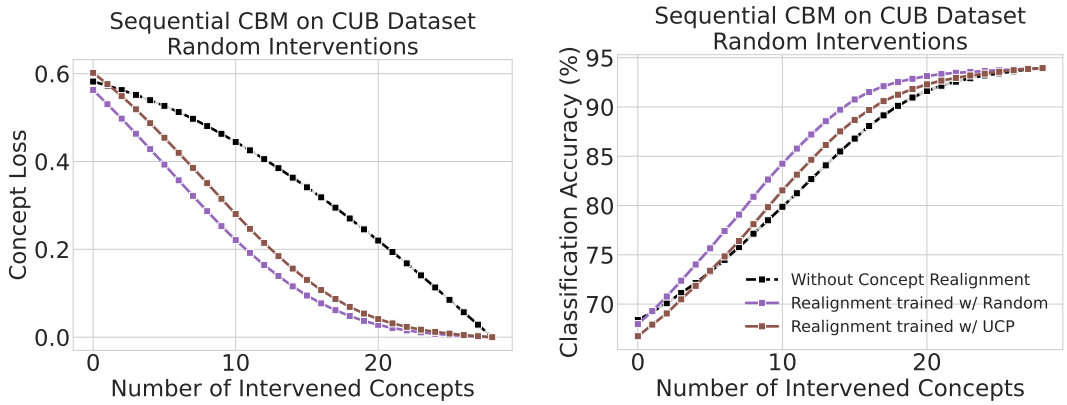


Figure 5.7: Concept prediction loss and classification accuracy under random interventions for realignment modules trained with random and UCP policy, respectively. Results indicate that alignment of policy used during training and deployment is important.

realignment module has been trained on the same random policy as well. This means that the realignment module adapts to the selection policy used during training. Thus to get the most benefits out of concept intervention realignment, selection policies should align during training and deployment.

Alignment b/w Realignment Module Components. Finally, we study how important the alignment between the concept realignment model and intervention policy (i.e., UCP) is to form the overall concept intervention realigment module. To accomplish this, we employ two module variations: (a) an *original policy* denoted as $\pi(\hat{c}_0)$, which only applies the UCP criterion to the original concept predictions generated by the base model without any concept realignment (i.e., the policy does not change over time), and (b) our default setup (*updated policy*), which informs the intervention policy using realigned concept values ($\pi(\kappa_t)$). Note that in both cases, the classification head still receives realigned concept embeddings, as we only want to study the importance of alignment between the concept realignment model and the intervention policy. Results in Fig. 5.8 clearly reveal that while simple realignment on its own can already help improve intervention efficacy, much larger efficacy gains are unlocked when both policy and the realignment model are utilized in conjunction.

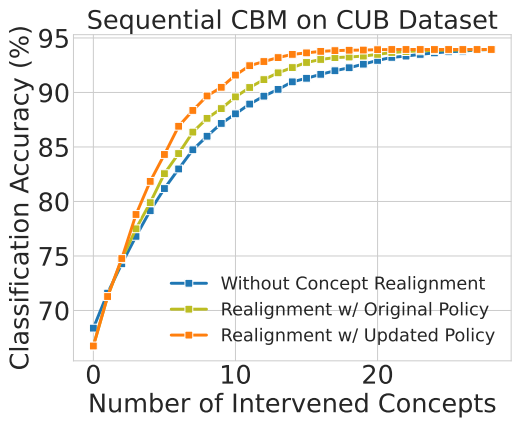


Figure 5.8: Classification accuracy vs concept interv. counts, showing our updated selection policies improving over the static one.

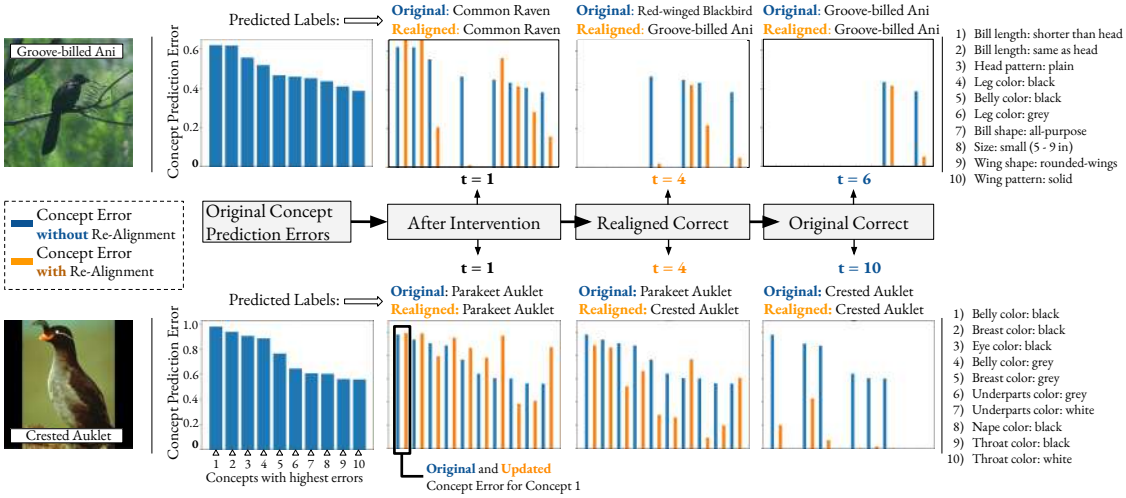


Figure 5.9: Qualitative examples for the improved intervention efficiency of CIRM. We show the change in concept prediction errors of the ten worst predicted concepts, as a function of concept intervention steps t. As can be seen, concept realignment allows concept error even for strongly mispredicted concepts to be significantly reduced with interventions, achieving correct label classification after much fewer interventions compared to a non-realigned baseline.

A Closer Look at Concept Realignment. To understand the impact of the realignment process qualitatively, we also provide examples in Fig. 5.9. In this figure, we showcase the impact of interventions on the top 10 concepts with the highest prediction errors, and the specific number of interventions required to predict the correct label. For both examples, we find that intervention on a single concept is insufficient to flip incorrect class predictions. However, as we intervene on more concepts, we can clearly see that concept realignment jointly allows concept prediction error - even on the initially worst predicted concepts - to be significantly reduced, while also reaching correct image classification with in parts less than half the number of interventions (for *Crested Auklet*). These results conceptually support the quantitative benefits of concept realignment seen in previous benchmark experiments.

5.5 Conclusion

In this work, we identify the independent treatment of concepts during test-time interventions in CBMs as a cause for reduced intervention efficacy. To remedy this problem, we propose a concept intervention realignment module - a simple and lightweight technique to automatically update concept assignments after human intervention on one or multiple concepts. Our experiments demonstrate significant gains in concept attribution as well as overall classification accuracy of concept-based models under intervention. We show that our approach is versatile and can be applied to a wide range of concept-based models, intervention policies, and training schemes. We believe that the reduction in required human interventions to reach performance targets facilitates the practical deployment of

concept-based models even in resource-constrained environments.

6

Thesis Discussion and Conclusion

6.1 Summary of Contributions

This thesis investigated how AI models can operate effectively under limited supervision. By focusing on zero-shot learning, data augmentation from few-shot data, and human intervention to interpretable models, the work aimed to improve the generalization of vision language models or vision models. Below, we summarize the main contributions across these three research areas.

6.1.1 Zero-shot Classification

Our contributions to zero-shot classification in vision-language models focused on thinking how foundation models can be leveraged effectively without additional training data:

- WaffleCLIP introduced a new perspective on how descriptor-based zero-shot classification should be interpreted. While prior work used a large language model to generate class descriptors to improve performance, we found that these gains could be attributed to an ensemble of multiple noisy prompts rather than meaningful semantics. It further suggested that by using both noisy prompts and semantic descriptors, the classification performance could be matched or improved.
- FLM tackled the problem of open-world compositional zero-shot learning, where the task is to classify novel combinations of known states and objects. By leveraging large language models to predict the feasibility of unseen state-object compositions, it removed the unfeasible classes from the class prediction set, leading to improved classification performance. This approach is beneficial when many candidate labels may be infeasible.

6.1.2 Data Augmentation by Generation with Few-shot Guidance

Our contributions to data augmentation by text-to-image generative models with few-shot guidance followed a trajectory beginning with the challenges of naive synthetic data generation method and moving towards two progressive solutions:

- We began by analyzing the limitations of synthetic data generated by text-to-image generative models in a standard way. We observed that synthetic images generated directly from class names often lack the object of interest or fail to capture fine-grained visual details, two factors that are critical for learning effective class-specific representations.
- DataDream addressed this issue by introducing a few-shot guided fine-tuning approach. Specifically, it fine-tuned the LoRA weights of a diffusion model on a small number of real class-specific images, enabling the model to generate more realistic and discriminative samples. This improved the alignment between synthetic and real distributions, leading to measurable gains in downstream classification performance when using those synthetic data as training data.
- Building on this, we proposed LoFT which extended the DataDream approach by fine-tuning LoRA modules per image and fusing them during inference. This allowed the model to combine class-relevant features from multiple real examples, resulting in improved diversity while maintaining the fidelity of the generated samples. The transition from DataDream to LoFT highlights the effectiveness of fine-tuning the diffusion model for synthetic data generation, especially in settings where only a few annotated images are available.

6.1.3 Human Intervention in Interpretable Models

Finally, we contributed to reducing the number of human intervention when interacting with the interpretable models:

- We introduced a Concept Intervention Realignment module for Concept Bottleneck Models (CBMs). CBMs offer a promising framework for interpretable AI, but their usefulness in practice depends on how efficiently users can intervene. Our proposed CRM module learned to realign related concepts following a minimal human correction, reducing the number of edits required to correct a model's reasoning. This improved the usability of human-in-the-loop systems, making interpretable models more practical for real-world deployment.

6.2 Broader Implications and Future Directions

Beyond the specific tasks addressed in this thesis, the research presented here offers broader insights into how foundation models can be adapted or leveraged in data-scarce scenarios.

6.2.1 Synthetic Data Generation Beyond Classification

While our proposed methods for synthetic data generation were developed in the context of image classification, the core idea could extend well beyond this task.

As foundation models grow in scale and complexity, they face persistent limitations such as hallucination, prompt sensitivity, poor adherence to textual instructions, and difficulties with fine-grained detail generation. These issues are evident not only in text-to-image generation models but also in large vision-language models and emerging text-to-video models.

In such cases, synthetic data generation could offer a promising way to address failure modes. For example, one can generate edge-case data in areas where models often fail. Suppose it is revealed that the large vision-language models (LVLMs) struggle with reasoning over visual charts. In that case, we can generate synthetic data with chart images, construct corresponding preference annotations, and use these to fine-tune the LVLMs through preference learning. In this way, targeted synthetic datasets can be curated to align model behavior better in the edge cases. Compared to real data collection, this approach is highly controllable, scalable, and cost-effective, making it well-suited for large-scale model alignment. Notably, data generation in these open-world tasks differs from traditional augmentation for classification: it is not bounded by predefined class labels or fixed taxonomies, but instead operates in open-ended, instruction-driven environments, where diversity and semantic alignment are critical.

6.2.2 Implicit Synthetic Data for Zero-Shot Learning in Generative Models

The principle of synthetic data generation has appeared explicitly/implicitly in other learning frameworks. For instance, self-rewarding language models is a language model itself being used to evaluate and score its own outputs and use them as training data for the next training iteration. The output of the LLMs in this case could be understood as synthetic data generation. In reinforcement learning, agents generate and interact with their own training environments through trial-and-error, which can be viewed as a form of synthetic data creation, with rewards serving as supervision.

These strategies highlight a common theme: models generating their own training signals, whether as outputs to be refined or as exploratory behaviors to be evaluated. Applying similar ideas to generative vision models, such as diffusion or video generative models, remains a relatively unexplored direction. Introducing structured, self-guided feedback loops for synthetic sample evaluation could help improve realism, relevance, and adherence to user instructions, thereby enhancing the generative process.

6.2.3 Less is More: Efficient and Informative Data Generation

One often-cited advantage of synthetic data is its scalability where it is computationally easier to scale the number of training data compared to collecting real data. However, a perhaps more compelling advantage is its controllability: the ability to generate data with specific properties, constraints, or content. This opens the door to the idea that less data, if well-targeted and information-rich, can be more valuable than large amounts of generic data. In other words, less is more.

This direction invites future work in synthetic data generation. Rather than flooding models with massive volumes of synthetic samples, we may aim to produce a minimal set of highly informative examples, tailored to known model weaknesses or aligned with desired behavior. This could lead to reductions not only in training time and storage, but also in the environmental cost associated with large-scale model training, an increasingly important concern in responsible AI development.

6.2.4 Collaborative and Teachable AI Systems

A key insight from this thesis is that interpretability should not be a passive, post-hoc feature, but an active part of the model's learning and adaptation process. Concept bottleneck models allow users to inspect and modify intermediate concept representations, offering a direct channel for human intervention. However, most current implementations rely on manually initiated corrections and lack feedback mechanisms from the model to the user. This limits their practicality in real-world applications, as users must manually check every concept to determine whether it aligns with their intent or domain knowledge.

One future direction is the development of interactive and proactive feedback mechanisms, where the model can initiate clarification requests, highlight uncertain or influential concepts, and adapt its behaviors based on the user's intervention. This vision of collaborative AI aligns with a broader shift toward systems that are not only trainable, but also teachable. Advancing this line of research could foster richer interaction between humans and models, with wide-reaching implications for both usability and trust in AI systems.

6.3 Final Reflections

Training models under limited supervision remains one of the challenges in AI research. This thesis has made substantial contributions toward addressing this problem by advancing model performance in zero-shot settings, proposing a few-shot-guided synthetic data generation framework, and introducing an efficient human intervention strategy through interpretable models. The trajectory of the thesis, from improving generalization in zero-shot classification (WaffleCLIP, FLM), to enabling scalable data generation with minimal supervision (DataDream, LoFT) and empowering users to minimally intervene in interpretable models (CRM) demonstrates how minimal supervision can be strategically leveraged for stronger generalization and usability.

We hope that this work encourages continued exploration at the intersection of zero-shot learning, synthetic data generation by few-shot guidance, and human-in-the-loop interpretability, bringing toward AI systems that are not only more data-efficient but also adaptive and collaborative with human intent.

Bibliography

[1] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. "Evaluation of output embeddings for fine-grained image classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2927–2936 (cit. on p. 2).

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2623–2631 (cit. on p. 72).

[3] Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. "Self-Consuming Generative Models Go MAD". In: *ICLR*. 2023 (cit. on pp. 38, 52).

[4] David Alvarez Melis and Tommi Jaakkola. "Towards robust interpretability with self-explaining neural networks". In: *Advances in neural information processing systems* 31 (2018) (cit. on p. 66).

[5] Rohan Anil et al. *PaLM 2 Technical Report*. 2023. eprint: arXivpreprintarXiv:2305.10403 (cit. on pp. 3, 25, 31).

[6] Anthropic. *Model Card and Evaluations for Claude Models*. 2023. URL: https://efficient-manatee.files.svdcdn.com/production/images/Model-Card-Claude-2.pdf (cit. on pp. 3, 25, 31).

[7] Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, and Christopher Re. "Ask Me Anything: A simple strategy for prompting language models". In: *ICLR*. 2023 (cit. on pp. 25, 33).

[8] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. "Synthetic data from diffusion models improves imagenet classification". In: *arXiv preprint arXiv:2304.08466* (2023) (cit. on pp. 36, 52).

[9] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. "Exploring visual prompts for adapting large-scale models". In: *arXiv:2203.17274* (2022) (cit. on pp. 3, 12).

[10] Hritik Bansal and Aditya Grover. "Leaving reality to imagination: Robust classification via generated datasets". In: *arXiv preprint arXiv:2302.02503* (2023) (cit. on pp. 38, 52).

[11] Yonatan Belinkov and Yonatan Bisk. "Synthetic and Natural Noise Both Break Neural Machine Translation". In: *ICLR*. 2018 (cit. on p. 12).

[12] Quentin Bertrand, Avishek Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. "On the Stability of Iterative Retraining of Generative Models on their own Data". In: *ICLR*. 2024 (cit. on p. 52).

[13] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. "Improving image generation with better captions". In: *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf* 2.3 (2023), p. 8 (cit. on pp. 38, 50, 52).

[14] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. "Paligemma: A versatile 3b vlm for transfer". In: *arXiv preprint arXiv:2407.07726* (2024) (cit. on pp. 55, 57).

[15] Moon Ye-Bin, Nam Hyeon-Woo, Wonseok Choi, Nayeong Kim, Suha Kwak, and Tae-Hyun Oh. "SYNAuG: Exploiting Synthetic Data for Data Imbalance Problems". In: *arXiv preprint arXiv:2308.00994* (2023) (cit. on p. 52).

[16] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006 (cit. on p. 2).

[17] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. "Food-101 – Mining Discriminative Components with Random Forests". In: *ECCV*. 2014 (cit. on pp. 16, 42, 44, 59).

[18] Alexander Brown, Nenad Tomasev, Jan Freyberg, Yuan Liu, Alan Karthikesalingam, and Jessica Schrouff. "Detecting shortcut learning for fair medical AI using shortcut testing". In: *Nature Communications* 14.1 (2023-07). ISSN: 2041-1723. DOI: 10.1038/s41467-023-39902-7. URL: http://dx.doi.org/10.1038/s41467-023-39902-7 (cit. on p. 64).

[19] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *NeurIPS* (2020) (cit. on pp. 2, 3, 10, 12, 15, 25, 31, 32).

[20] Vanessa Buhrmester, David Münch, and Michael Arens. *Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey*. 2019. arXiv: 1911.12116 [cs.AI] (cit. on p. 65).

[21] Sebastian Bujwid and Josephine Sullivan. "Large-Scale Zero-Shot Image Classification from Rich and Diverse Textual Descriptions". In: *Workshop on Beyond Vision and LANguage: inTEgrating Real-world kNowledge*. 2021 (cit. on p. 12).

[22] Max F Burg, Florian Wenzel, Dominik Zietlow, Max Horn, Osama Makansi, Francesco Locatello, and Chris Russell. "Image retrieval outperforms diffusion models on data augmentation". In: *TMLR* (2023) (cit. on p. 38).

[23] Stephen Casper, Carson Ezell, Charlotte Siegmann, Noam Kolt, Taylor Lynn Curtis, Benjamin Bucknall, Andreas Haupt, Kevin Wei, Jérémy Scheurer, Marius Hobbhahn, Lee Sharkey, Satyapriya Krishna, Marvin Von Hagen, Silas Alberti, Alan Chan, Qinyi Sun, Michael Gerovitch, David Bau, Max Tegmark, David Krueger, and Dylan Hadfield-Menell. *Black-Box Access is Insufficient for Rigorous AI Audits*. 2024. arXiv: 2401.14446 [cs.CY] (cit. on pp. 64, 65).

[24] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild". In: *ECCV*. 2016 (cit. on p. 25).

[25] Kushal Chauhan, Rishabh Tiwari, Jan Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. "Interactive concept bottleneck models". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 5. 2023, pp. 5948–5955 (cit. on p. 67).

[26] Aochuan Chen, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, and Sijia Liu. "Understanding and Improving Visual Prompting: A Label-Mapping Perspective". In: *arXiv:2211.11635* (2022) (cit. on p. 12).

[27] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. "PLOT: Prompt Learning with Optimal Transport for Vision-Language Models". In: *ICLR*. 2023 (cit. on p. 10).

[28] Jiaao Chen, Zichao Yang, and Diyi Yang. "MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification". In: *ACL*. 2020 (cit. on p. 12).

[29] Xinlei Chen, Saining Xie, and Kaiming He. "An empirical study of training self-supervised vision transformers". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9640–9649 (cit. on p. 59).

[30] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 24185–24198 (cit. on p. 3).

[31] Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. "Towards Robust Neural Machine Translation". In: *ACL*. 2018 (cit. on p. 12).

[32] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. 2023. URL: https://lmsys.org/blog/2023-03-30-vicuna/ (cit. on pp. 3, 25, 28, 31, 32).

[33] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. "Ilvr: Conditioning method for denoising diffusion probabilistic models". In: *arXiv preprint arXiv:2108.02938* (2021) (cit. on p. 38).

[34] Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. "The Curious Layperson: Fine-Grained Image Recognition without Expert Labels". In: *BMVC*. 2021 (cit. on p. 12).

[35] Alexandra Chouldechova. *Fair prediction with disparate impact: A study of bias in recidivism prediction instruments*. 2016. arXiv: 1610.07524 [stat.AP] (cit. on p. 64).

[36] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. "Describing Textures in the Wild". In: *CVPR*. 2014 (cit. on pp. 16, 42, 59, 118).

[37] Victor G. Turrisi da Costa, Nicola Dall'Asen, Yiming Wang, Nicu Sebe, and Elisa Ricci. *Diversified in-domain synthesis with efficient fine-tuning for few-shot classification*. 2023. arXiv: 2312.03046 [cs.CV] (cit. on pp. 35–37, 39, 41–45, 48, 51, 52, 58, 116, 119).

[38] Timothee Cour, Ben Sapp, and Ben Taskar. "Learning from partial labels". In: *The Journal of Machine Learning Research* 12 (2011), pp. 1501–1536 (cit. on p. 2).

[39] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. "Autoaugment: Learning augmentation strategies from data". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 113–123 (cit. on p. 2).

[40] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. "Hyperspherical Variational Auto-Encoders". In: *Conference on Uncertainty in Artificial Intelligence* (2018) (cit. on p. 21).

[41] Ofer Dekel and Ohad Shamir. "Multiclass-multilabel classification with more classes than examples". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 137–144 (cit. on p. 2).

[42] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *CVPR*. 2009 (cit. on pp. 2, 16, 36, 42, 57).

[43] Terrance DeVries and Graham W Taylor. "Improved regularization of convolutional neural networks with cutout". In: *arXiv:1708.04552* (2017) (cit. on p. 2).

[44] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. "A survey on in-context learning". In: *arXiv preprint arXiv:2301.00234* (2022) (cit. on p. 3).

[45] Amil Dravid, Yossi Gandelsman, Kuan-Chieh Wang, Rameen Abdal, Gordon Wetzstein, Alexei A Efros, and Kfir Aberman. "Interpreting the Weight Space of Customized Diffusion Models". In: *arXiv preprint arXiv:2406.09413* (2024) (cit. on p. 52).

[46] Xuefeng Du, Yiyou Sun, Jerry Zhu, and Yixuan Li. "Dream the impossible: Outlier imagination with diffusion models". In: *NeurIPS* (2023) (cit. on p. 52).

[47] Natalie Dullerud, Karsten Roth, Kimia Hamidieh, Nicolas Papernot, and Marzyeh Ghassemi. "Is Fairness Only Metric Deep? Evaluating and Addressing Subgroup Gaps in Deep Metric Learning". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=js62_xuLDDv (cit. on p. 64).

[48] Lisa Dunlap, Clara Mohri, Devin Guillory, Han Zhang, Trevor Darrell, Joseph E. Gonzalez, Aditi Raghunathan, and Anja Rohrbach. *Using Language to Extend to Unseen Domains*. 2023. arXiv: 2210.09520 [cs.CV] (cit. on p. 12).

[49] Lisa Dunlap, Alyssa Umino, Han Zhang, Jiezhi Yang, Joseph E. Gonzalez, and Trevor Darrell. *Diversify Your Vision Datasets with Automatic Diffusion-Based Augmentation*. 2023. arXiv: 2305.16289 [cs.CV] (cit. on pp. 36, 38, 50, 52, 53, 55).

[50] Juan Manuel Durán and Karin Rolanda Jongsma. "Who is afraid of black box algorithms? On the epistemological and ethical basis of trust in medical AI". In: *Journal of Medical Ethics* 47.5 (2021), pp. 329–335. ISSN: 0306-6800. DOI: 10.1136/medethics-2020-106820. eprint: https://jme.bmj.com/content/47/5/329.full.pdf. URL: https://jme.bmj.com/content/47/5/329 (cit. on p. 65).

[51] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. "Fairness through awareness". In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ITCS '12. Cambridge, Massachusetts: Association for Computing Machinery, 2012, pp. 214–226. ISBN: 9781450311151. DOI: 10.1145/2090236.2090255. URL: https://doi.org/10.1145/2090236.2090255 (cit. on p. 64).

[52] Mohamed Elhoseiny, Yizhe Zhu, Han Zhang, and Ahmed Elgammal. "Link the head to the" beak": Zero shot learning from noisy text description at part precision". In: *CVPR*. 2017 (cit. on p. 12).

[53] EUGDPR. *GDPR. General data protection regulation*. 2017 (cit. on p. 65).

[54] Elias Eulig, Piyapat Saranrittichai, Chaithanya Kumar Mummadi, Kilian Rambach, William Beluch, Xiahan Shi, and Volker Fischer. "DiagViB-6: A Diagnostic Benchmark Suite for Vision Models in the Presence of Shortcut and Generalization Opportunities". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021-10 (cit. on p. 64).

[55] Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. "Scaling laws of synthetic images for model training... for now". In: *CVPR*. 2024 (cit. on pp. 50, 51, 53–55, 60).

[56] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. "A Survey of Data Augmentation Approaches for NLP". In: *ACL-IJCNLP*. 2021 (cit. on p. 12).

[57] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks". In: *International conference on machine learning*. PMLR. 2017, pp. 1126–1135 (cit. on p. 2).

[58] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. "Devise: A deep visual-semantic embedding model". In: *Advances in neural information processing systems* 26 (2013) (cit. on p. 2).

[59] Yunxiang Fu, Chaoqi Chen, Yu Qiao, and Yizhou Yu. "Dreamda: Generative data augmentation with diffusion models". In: *arXiv preprint arXiv:2403.12803* (2024) (cit. on p. 52).

[60] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. "An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion". In: *ICLR*. 2022 (cit. on pp. 37, 38, 52, 53, 62).

[61] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. "Clip-adapter: Better vision-language models with feature adapters". In: *International Journal of Computer Vision* 132.2 (2024), pp. 581–595 (cit. on p. 3).

[62] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix Wichmann. "Shortcut learning in deep neural networks". In: *Nature Machine Intelligence* 2 (2020), pp. 665–673 (cit. on p. 64).

[63] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014) (cit. on p. 38).

[64] Yanming Guo, Yu Liu, Erwin M Bakker, Yuanhao Guo, and Michael S Lew. "CNN-RNN: a large-scale hierarchical image classification framework". In: *Multimedia tools and applications* (2018) (cit. on p. 15).

[65] Hasan Abed Al Kader Hammoud, Hani Itani, Fabio Pizzati, Philip Torr, Adel Bibi, and Bernard Ghanem. "SynthCLIP: Are We Ready for a Fully Synthetic CLIP Training?" In: *arXiv preprint arXiv:2402.01832* (2024) (cit. on pp. 38, 52, 53).

[66] Xiaoshuai Hao, Yi Zhu, Srikar Appalaraju, Aston Zhang, Wanqian Zhang, Bo Li, and Mu Li. "Mixgen: A new multi-modal data augmentation". In: *WACV*. 2023 (cit. on p. 12).

[67] Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. "Addressing leakage in concept bottleneck models". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 23386–23397 (cit. on p. 66).

[68] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *CVPR*. 2016 (cit. on pp. 58, 122).

[69] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and XIAOJUAN QI. "Is synthetic data from generative models ready for image recognition?" In: *ICLR*. 2023 (cit. on pp. 12, 35–39, 42–45, 50–54, 116, 119).

[70] Xiangteng He and Yuxin Peng. "Fine-grained image classification via combining vision and language". In: *CVPR*. 2017 (cit. on p. 12).

[71] Georg Heigold, Stalin Varanasi, Günter Neumann, and Josef van Genabith. "How Robust Are Character-Based Word Embeddings in Tagging and MT Against Wrod Scramlbing or Randdm Nouse?" In: *Association for Machine Translation in the Americas*. 2018 (cit. on p. 12).

[72] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. "EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification." In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2017). URL: http://dblp.uni-trier.de/db/journals/corr/corr1709.html#abs-1709-00029 (cit. on pp. 16, 42, 59).

[73] Reyhane Askari Hemmat, Mohammad Pezeshki, Florian Bordes, Michal Drozdzal, and Adriana Romero-Soriano. "Feedback-guided data synthesis for imbalanced classification". In: *arXiv preprint arXiv:2310.00158* (2023) (cit. on p. 52).

[74] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. "The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization". In: *ICCV* (2021) (cit. on pp. 19, 20).

[75] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. "Natural Adversarial Examples". In: *CVPR* (2021) (cit. on pp. 19, 20).

[76] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "Gans trained by a two time-scale update rule converge to a local nash equilibrium". In: *Advances in neural information processing systems* 30 (2017) (cit. on pp. 46, 124).

[77] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *NeurIPS* (2020) (cit. on p. 50).

[78] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780 (cit. on pp. 71, 76).

[79] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. "Meta-learning in neural networks: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* 44.9 (2021), pp. 5149–5169 (cit. on p. 2).

[80] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *ICLR*. 2021 (cit. on pp. 37, 39–42, 52, 55).

[81] Tony Huang, Jack Chu, and Fangyun Wei. "Unsupervised prompt learning for vision-language models". In: *arXiv:2204.03649* (2022) (cit. on p. 10).

[82] Phillip Isola, Joseph J Lim, and Edward H Adelson. "Discovering states and transformations in image collections". In: *CVPR*. 2015 (cit. on pp. 24, 28, 112).

[83] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. "Few-shot learning with retrieval augmented language models". In: *arXiv preprint arXiv:2208.03299* 1.2 (2022), p. 4 (cit. on p. 3).

[84] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. "Scaling up visual and vision-language representation learning with noisy text supervision". In: *ICML*. 2021 (cit. on p. 3).

[85] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. "Visual prompt tuning". In: *ECCV*. 2022 (cit. on pp. 43, 44).

[86] Shyamgopal Karthik, Massimiliano Mancini, and Zeynep Akata. "Kg-sp: Knowledge guided simple primitives for open world compositional zero-shot learning". In: *CVPR*. 2022 (cit. on pp. 23, 25, 26, 28).

[87] Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. "Probabilistic Concept Bottleneck Models". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023-23–29 Jul, pp. 16521–16540. URL: https://proceedings.mlr.press/v202/kim23g.html (cit. on p. 66).

[88] Jae Myung Kim, Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. "Feasibility with Language Models for Open-World Compositional Zero-Shot Learning". In: *CVPR Workshop on OODCV* (2024) (cit. on p. 136).

[89] Jae Myung Kim, Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. "LoFT: LoRA-Fused Training Dataset Generation with Few-shot Guidance". In: *Under Review* (2025) (cit. on p. 136).

[90] Jae Myung Kim, Jessica Bader, Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. "DataDream: Few-shot Guided Dataset Generation". In: *ECCV* (2024) (cit. on pp. 51, 52, 54, 57, 58, 136).

[91] Jae Myung Kim, A Koepke, Cordelia Schmid, and Zeynep Akata. "Exposing and mitigating spurious correlations for cross-modal retrieval". In: *CVPR*. 2023 (cit. on p. 137).

[92] Youngwook Kim, Jae Myung Kim, Zeynep Akata, and Jungwoo Lee. "Large loss matters in weakly supervised multi-label classification". In: *CVPR*. 2022 (cit. on p. 136).

[93] Youngwook Kim, Jae Myung Kim, Jieun Jeong, Cordelia Schmid, Zeynep Akata, and Jungwoo Lee. "Bridging the gap between model explanations in partially annotated multi-label classification". In: *CVPR*. 2023 (cit. on p. 137).

[94] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013) (cit. on p. 38).

[95] Michael Kirchhof, Karsten Roth, Zeynep Akata, and Enkelejda Kasneci. "A Non-isotropic Probabilistic Take on Proxy-based Deep Metric Learning". In: *ECCV*. 2022 (cit. on p. 21).

[96] Sosuke Kobayashi. "Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations". In: *NAACL-HLT*. 2018 (cit. on p. 12).

[97] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. "Concept bottleneck models". In: *International conference on machine learning*. PMLR. 2020, pp. 5338–5348 (cit. on pp. 4, 64–67, 70, 72, 75).

[98] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. "Large language models are zero-shot reasoners". In: *Advances in neural information processing systems* 35 (2022), pp. 22199–22213 (cit. on p. 3).

[99] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. "Do Better ImageNet Models Transfer Better?" In: *CVPR*. 2019 (cit. on p. 16).

[100] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. "3d object representations for fine-grained categorization". In: *ICCV workshop*. 2013, pp. 554–561 (cit. on pp. 16, 20, 42–44, 59, 118–120).

[101] Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. "Can language models learn from explanations in context?" In: *EMNLP*. 2022 (cit. on p. 25).

[102] Shiye Lei, Hao Chen, Sen Zhang, Bo Zhao, and Dacheng Tao. "Image captions are natural prompts for text-to-image models". In: *arXiv preprint arXiv:2307.08526* (2023) (cit. on p. 52).

[103] David D Lewis and Jason Catlett. "Heterogeneous uncertainty sampling for supervised learning". In: *Machine learning proceedings 1994*. Elsevier, 1994, pp. 148–156 (cit. on p. 68).

[104] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks". In: *Advances in neural information processing systems* 33 (2020), pp. 9459–9474 (cit. on p. 3).

[105] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. *Caltech 101*. 2022-04. DOI: 10.22002/D1.20086 (cit. on pp. 42, 44, 59).

[106] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models". In: *arXiv:2301.12597* (2023) (cit. on p. 3).

[107] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation". In: *ICML*. 2022 (cit. on p. 3).

[108] Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. "Symmetry and group in attribute-object compositions". In: *CVPR*. 2020 (cit. on p. 24).

[109] Shaobo Lin, Kun Wang, Xingyu Zeng, and Rui Zhao. "Explore the Power of Synthetic Data on Few-shot Object Detection". In: *CVPR*. 2023 (cit. on pp. 35, 38, 52, 53).

[110] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. "Visual Instruction Tuning". In: *NeurIPS*. 2023 (cit. on pp. 2, 3, 116).

[111] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. "What Makes Good In-Context Examples for GPT-3?" In: *arXiv preprint arXiv:2101.06804* (2021) (cit. on p. 3).

[112] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing". In: *ACM Computing Surveys* (2023) (cit. on p. 10).

[113] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. "More control for free! image synthesis with semantic diffusion guidance". In: *WACV*. 2023 (cit. on p. 38).

[114] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. "Mmbench: Is your multi-modal model an all-around player?" In: *European conference on computer vision*. Springer. 2024, pp. 216–233 (cit. on p. 3).

[115] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. "Deep learning face attributes in the wild". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 3730–3738 (cit. on pp. 66, 72).

[116] Francesco Locatello, Gabriele Abbati, Tom Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. "On the fairness of disentangled representations". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019 (cit. on p. 65).

[117] Jochem Loedeman, Maarten C Stol, Tengda Han, and Yuki M Asano. "Prompt Generation Networks for Efficient Adaptation of Frozen Vision Transformers". In: *arXiv:2210.06466* (2022) (cit. on p. 12).

[118] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *ICLR*. 2018 (cit. on pp. 42, 57, 116).

[119] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. "Prompt distribution learning". In: *CVPR*. 2022 (cit. on p. 12).

[120] Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. "Promises and pitfalls of black-box concept learning models". In: *arXiv preprint arXiv:2106.13314* (2021) (cit. on p. 66).

[121] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. *Fine-Grained Visual Classification of Aircraft*. Tech. rep. 2013. arXiv: 1306.5151 [cs-cv] (cit. on pp. 37, 42–45, 48, 59, 117, 119).

[122] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. "Fine-grained visual classification of aircraft". In: *arXiv:1306.5151* (2013) (cit. on pp. 16, 20).

[123] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. "Open world compositional zero-shot learning". In: *CVPR*. 2021 (cit. on pp. 23, 25–28).

[124] Chengzhi Mao, Revant Teotia, Amrutha Sundar, Sachit Menon, Junfeng Yang, Xin Wang, and Carl Vondrick. *Doubly Right Object Recognition: A Why Prompt for Visual Rationales*. 2023. arXiv: 2212.06202 [cs.CV] (cit. on p. 12).

[125] Emanuele Marconato, Andrea Passerini, and Stefano Teso. "Glancenets: Interpretable, leak-proof concept-based models". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 21212–21227 (cit. on p. 66).

[126] Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. "Do concept bottleneck models learn as intended?" In: *arXiv preprint arXiv:2105.04289* (2021) (cit. on p. 66).

[127] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. *A Survey on Bias and Fairness in Machine Learning*. 2022. arXiv: 1908.09635 [cs.LG] (cit. on p. 64).

[128] Sachit Menon and Carl Vondrick. "Visual Classification via Description from Large Language Models". In: *ICLR*. 2023 (cit. on pp. 3, 10–17, 19, 20, 106–108).

[129] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998 (cit. on p. 12).

[130] Ishan Misra, Abhinav Gupta, and Martial Hebert. "From red wine to red tomato: Composition with context". In: *CVPR*. 2017 (cit. on p. 24).

[131] Muhammad Ferjad Naeem, Muhammad Gul Zain Ali Khan, Yongqin Xian, Muhammad Zeshan Afzal, Didier Stricker, Luc Van Gool, and Federico Tombari. "I2MVFormer: Large Language Model Generated Multi-View Document Supervision for Zero-Shot Image Classification". In: *CVPR*. 2023 (cit. on p. 12).

[132] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. "Learning graph embeddings for compositional zero-shot learning". In: *CVPR*. 2021 (cit. on pp. 24, 28, 112).

[133] Muhammad Ferjad Naeem, Yongqin Xian, Luc Van Gool, and Federico Tombari. "I2dformer: Learning image to document attention for zero-shot image classification". In: *NeurIPS*. 2022 (cit. on p. 12).

[134] Tushar Nagarajan and Kristen Grauman. "Attributes as operators: factorizing unseen attribute-object compositions". In: *ECCV*. 2018 (cit. on p. 24).

[135] Nihal V. Nayak, Peilin Yu, and Stephen Bach. "Learning to Compose Soft Prompts for Compositional Zero-Shot Learning". In: *ICLR*. 2023 (cit. on pp. 24, 28, 29, 112, 113).

[136] Alex Nichol, Joshua Achiam, and John Schulman. "On first-order meta-learning algorithms". In: *arXiv preprint arXiv:1803.02999* (2018) (cit. on p. 2).

[137] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. "Glide: Towards photorealistic image generation and editing with text-guided diffusion models". In: *arXiv preprint arXiv:2112.10741* (2021) (cit. on pp. 38, 43, 52, 116).

[138] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. "Text classification from labeled and unlabeled documents using EM". In: *Machine learning* 39 (2000), pp. 103–134 (cit. on p. 2).

[139] Maria-Elena Nilsback and Andrew Zisserman. "Automated Flower Classification over a Large Number of Classes". In: *Indian Conference on Computer Vision, Graphics and Image Processing*. 2008 (cit. on pp. 16, 20, 42, 43, 59, 118, 121).

[140] Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. "Evaluating Robustness to Input Perturbations for Neural Machine Translation". In: *ACL*. 2020 (cit. on p. 12).

[141] Zachary Novack, Saurabh Garg, Julian McAuley, and Zachary C Lipton. "Chils: Zero-shot image classification with hierarchical label sets". In: *arXiv:2302.02551* (2023) (cit. on pp. 3, 10–12, 19).

[142] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. "Label-free Concept Bottleneck Models". In: *The Eleventh International Conference on Learning Representations*. 2022 (cit. on p. 66).

[143] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 (cit. on pp. 3, 31).

[144] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. "Training language models to follow instructions with human feedback". In: *NeurIPS* (2022) (cit. on pp. 25, 31, 32).

[145] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. "Cats and dogs". In: *CVPR*. 2012 (cit. on pp. 16, 42, 44, 59).

[146] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *NeurIPS*. 2019 (cit. on pp. 16, 72).

[147] Tzuf Paz-Argaman, Reut Tsarfaty, Gal Chechik, and Yuval Atzmon. "ZEST: Zero-shot Learning from Text Descriptions using Textual Similarity and Visual Summarization". In: *EMNLP*. 2020 (cit. on p. 12).

[148] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *EMNLP*. 2014 (cit. on pp. 23, 28).

[149] Pablo Pernias, Dominic Rampas, Mats Leon Richter, Christopher Pal, and Marc Aubreville. "Würstchen: An Efficient Architecture for Large-Scale Text-to-Image Diffusion Models". In: *ICLR*. 2023 (cit. on p. 38).

[150] Samuele Lo Piano. "Ethical principles in machine learning and artificial intelligence: cases from the field and possible ways forward". In: *Palgrave Communications* 7.1 (2020), pp. 1–7. URL: https://EconPapers.repec.org/RePEc:pal:palcom:v:7:y:2020:i:1:d:10.1057_s41599-020-0501-9 (cit. on p. 65).

[151] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. "SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis". In: *ICLR*. 2023 (cit. on pp. 38, 50, 52).

[152] Sarah Pratt, Rosanne Liu, and Ali Farhadi. "What does a platypus look like? Generating customized prompts for zero-shot image classification". In: *arXiv:2209.03320* (2022) (cit. on pp. 3, 10–13, 19, 22).

[153] Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. "Task-driven modular networks for zero-shot compositional learning". In: *ICCV*. 2019, pp. 3593–3602 (cit. on pp. 24, 25, 28).

[154] Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. "Controlling text-to-image diffusion by orthogonal finetuning". In: *NeurIPS* (2023) (cit. on p. 52).

[155] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. "Learning transferable visual models from natural language supervision". In: *ICML*. 2021 (cit. on pp. 2, 3, 10, 12, 13, 15–17, 19, 20, 24, 29, 41–43, 57, 58, 107, 113).

[156] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. "Self-taught learning: transfer learning from unlabeled data". In: *Proceedings of the 24th international conference on Machine learning*. 2007, pp. 759–766 (cit. on p. 2).

[157] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. "Hierarchical text-conditional image generation with clip latents". In: *arXiv preprint arXiv:2204.06125* 1.2 (2022), p. 3 (cit. on pp. 38, 50, 52).

[158] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. "Learning deep representations of fine-grained visual descriptions". In: *CVPR*. 2016 (cit. on p. 12).

[159] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-Resolution Image Synthesis With Latent Diffusion Models". In: *CVPR*. 2022 (cit. on pp. 5, 35, 37–39, 42, 50, 52, 54, 56, 116).

[160] Bernardino Romera-Paredes and Philip Torr. "An embarrassingly simple approach to zero-shot learning". In: *International conference on machine learning*. PMLR. 2015, pp. 2152–2161 (cit. on p. 2).

[161] Karsten Roth, Mark Ibrahim, Zeynep Akata, Pascal Vincent, and Diane Bouchacourt. "Disentanglement of Correlated Factors via Hausdorff Factorized Support". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=0KcJhpQiGiX (cit. on p. 65).

[162] Karsten Roth, Jae Myung Kim, A Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. "Waffling around for performance: Visual classification with random words and broad concepts". In: *ICCV*. 2023 (cit. on pp. 3, 136).

[163] Karsten Roth, Lukas Thede, A. Sophia Koepke, Oriol Vinyals, Olivier J Henaff, and Zeynep Akata. "Fantastic Gains and Where to Find Them: On the Existence and Prospect of General Knowledge Transfer between Any Pretrained Model". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=m50eKHCttz (cit. on p. 64).

[164] Karsten Roth, Oriol Vinyals, and Zeynep Akata. "Integrating Language Guidance Into Vision-Based Deep Metric Learning". In: *CVPR*. 2022 (cit. on p. 12).

[165] Karsten Roth, Oriol Vinyals, and Zeynep Akata. "Non-isotropy Regularization for Proxy-based Deep Metric Learning". In: *CVPR*. 2022 (cit. on p. 21).

[166] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. "Learning to retrieve prompts for in-context learning". In: *arXiv preprint arXiv:2112.08633* (2021) (cit. on p. 3).

[167] Frank Ruis, Gertjan Burghouts, and Doina Bucur. "Independent prototype propagation for zero-shot compositionality". In: *NeurIPS* (2021) (cit. on p. 24).

[168] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation". In: *CVPR*. 2023, pp. 22500–22510 (cit. on pp. 37, 38, 41, 52, 117).

[169] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Wei Wei, Tingbo Hou, Yael Pritch, Neal Wadhwa, Michael Rubinstein, and Kfir Aberman. "Hyperdreambooth: Hypernetworks for fast personalization of text-to-image models". In: *CVPR*. 2024 (cit. on p. 52).

[170] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. "Photorealistic text-to-image diffusion models with deep language understanding". In: *NeurIPS* (2022) (cit. on pp. 38, 50, 52).

[171] Gözde Gül Şahin. "To Augment or Not to Augment? A Comparative Study on Text Augmentation Techniques for Low-Resource NLP". In: *Computational Linguistics* (2022) (cit. on p. 12).

[172] Leonard Salewski, Stephan Alaniz, Isabel Rio-Torto, Eric Schulz, and Zeynep Akata. "In-Context Impersonation Reveals Large Language Models' Strengths and Biases". In: *NeurIPS*. 2023 (cit. on p. 27).

[173] Mert Bulent Sariyildiz, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. "Fake it till you make it: Learning transferable representations from synthetic ImageNet clones". In: *CVPR*. 2023 (cit. on pp. 36, 38, 47, 50, 52–54).

[174] Yoshihide Sawada and Keigo Nakamura. "Concept bottleneck model with additional unsupervised concepts". In: *IEEE Access* 10 (2022), pp. 41758–41765 (cit. on p. 66).

[175] Madeline Schiappa, Raiyaan Abdullah, Shehreen Azad, Jared Claypoole, Michael Cogswell, Ajay Divakaran, and Yogesh Rawat. "Probing conceptual understanding of large visual-language models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 1797–1807 (cit. on p. 3).

[176] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002 (cit. on p. 2).

[177] Burr Settles. "Active learning literature survey". In: (2009) (cit. on p. 2).

[178] Sahand Sharifzadeh, Christos Kaplanis, Shreya Pathak, Dharshan Kumaran, Anastasija Ilic, Jovana Mitrovic, Charles Blundell, and Andrea Banino. "Synth2: Boosting Visual-Language Models with Synthetic Captions and Image Embeddings". In: *arXiv preprint arXiv:2403.07750* (2024) (cit. on p. 52).

[179] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004 (cit. on p. 2).

[180] Sheng Shen, Chunyuan Li, Xiaowei Hu, Yujia Xie, Jianwei Yang, Pengchuan Zhang, Anna Rohrbach, Zhe Gan, Lijuan Wang, Lu Yuan, et al. "K-lite: Learning transferable visual models with external knowledge". In: *arXiv:2204.09222* (2022) (cit. on p. 12).

[181] Ivaxi Sheth, Aamer Abdul Rahman, Laya Rafiee Sevyeri, Mohammad Havaei, and Samira Ebrahimi Kahou. "Learning from uncertain concepts via test time interventions". In: *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022*. 2022 (cit. on p. 67).

[182] Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. "Instantbooth: Personalized text-to-image generation without test-time finetuning". In: *CVPR*. 2024 (cit. on p. 52).

[183] Joonghyuk Shin, Minguk Kang, and Jaesik Park. "Fill-up: Balancing long-tailed data with generative models". In: *arXiv preprint arXiv:2306.07200* (2023) (cit. on p. 52).

[184] Sungbin Shin, Yohan Jo, Sungsoo Ahn, and Namhoon Lee. "A Closer Look at the Intervention Procedure of Concept Bottleneck Models". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023-23–29 Jul, pp. 31504–31520. URL: https://proceedings.mlr.press/v202/shin23a.html (cit. on pp. 65, 67, 68, 70).

[185] Jordan Shipard, Arnold Wiliem, Kien Nguyen Thanh, Wei Xiang, and Clinton Fookes. *Diversity is Definitely Needed: Improving Model-Agnostic Zero-shot Classification via Stable Diffusion*. 2023. arXiv: 2302.03298 [cs.CV] (cit. on pp. 36, 38, 50, 52, 53).

[186] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". In: *Journal of big data* 6.1 (2019), pp. 1–48 (cit. on pp. 2, 12).

[187] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. "Test-time prompt tuning for zero-shot generalization in vision-language models". In: *arXiv:2209.07511* (2022) (cit. on p. 12).

[188] Ravid Shwartz-Ziv and Naftali Tishby. *Opening the Black Box of Deep Neural Networks via Information*. 2017. arXiv: 1703.00810 [cs.LG] (cit. on p. 65).

[189] Nishad Singhi, Jae Myung Kim, Karsten Roth, and Zeynep Akata. "Improving intervention efficacy via concept realignment in concept bottleneck models". In: *ECCV*. 2024 (cit. on pp. 4, 136).

[190] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 2).

[191] Robyn Speer, Joshua Chin, and Catherine Havasi. "Conceptnet 5.5: An open multilingual graph of general knowledge". In: *AAAI*. 2017 (cit. on pp. 23, 28).

[192] Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, S Yu Philip, and Lifang He. "Mixup-Transformer: Dynamic Data Augmentation for NLP Tasks". In: *Computational Linguistics*. 2020 (cit. on p. 12).

[193] Zhaorui Tan, Xi Yang, and Kaizhu Huang. "Semantic-aware data augmentation for text-to-image synthesis". In: *AAAI*. 2024 (cit. on p. 52).

[194] Yuwei Tang, Zhenyi Lin, Qilong Wang, Pengfei Zhu, and Qinghua Hu. "Amu-tuning: Effective logit bias for clip-based few-shot learning". In: *CVPR*. 2024 (cit. on pp. 3, 57, 59).

[195] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. "Winoground: Probing vision and language models for visio-linguistic compositionality". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5238–5248 (cit. on p. 3).

[196] Yonglong Tian, Lijie Fan, Kaifeng Chen, Dina Katabi, Dilip Krishnan, and Phillip Isola. "Learning vision from models rivals learning vision from data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 15887–15898 (cit. on p. 52).

[197] Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. "Stablerep: Synthetic images from text-to-image models make strong visual representation learners". In: *NeurIPS* (2023) (cit. on pp. 35, 38, 52).

[198] Michael E Tipping. "Sparse Bayesian learning and the relevance vector machine". In: *Journal of machine learning research* 1.Jun (2001), pp. 211–244 (cit. on p. 2).

[199] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971* (2023) (cit. on pp. 3, 25).

[200] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. "Llama 2: Open foundation and fine-tuned chat models". In: *arXiv preprint arXiv:2307.09288* (2023) (cit. on pp. 3, 25, 31).

[201] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. "SuS-X: Training-Free Name-Only Transfer of Vision-Language Models". In: *arXiv:2211.16198* (2022) (cit. on pp. 11, 12).

[202] Uddeshya Upadhyay, Jae Myung Kim, Cordelia Schmidt, Bernhard Schölkopf, and Zeynep Akata. "Likelihood annealing: Fast calibrated uncertainty for regression". In: *arXiv preprint arXiv:2302.11012* (2023) (cit. on p. 137).

[203] Thanh Van Le, Hao Phung, Thuan Hoang Nguyen, Quan Dao, Ngoc N Tran, and Anh Tran. "Anti-dreambooth: Protecting users from personalized text-to-image synthesis". In: *ICCV*. 2023 (cit. on p. 52).

[204] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999 (cit. on p. 2).

[205] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. "Matching networks for one shot learning". In: *Advances in neural information processing systems* 29 (2016) (cit. on p. 2).

[206] Sandra Wachter, Brent Mittelstadt, and Chris Russell. *Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR*. 2018. arXiv: 1711.00399 [cs.AI] (cit. on p. 65).

[207] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. "The caltech-ucsd birds-200-2011 dataset". In: (2011) (cit. on pp. 16, 65, 66, 72).

[208] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. "Learning Robust Global Representations by Penalizing Local Predictive Power". In: *Advances in Neural Information Processing Systems*. 2019, pp. 10506–10518 (cit. on pp. 19, 20).

[209] Tongzhou Wang and Phillip Isola. "Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere". In: *ICML*. 2020 (cit. on p. 21).

[210] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. "Generalizing from a few examples: A survey on few-shot learning". In: *ACM computing surveys (csur)* 53.3 (2020), pp. 1–34 (cit. on p. 2).

[211] Zhengbo Wang, Jian Liang, Lijun Sheng, Ran He, Zilei Wang, and Tieniu Tan. "A Hard-to-Beat Baseline for Training-free CLIP-based Adaptation". In: *ICLR*. 2024 (cit. on p. 116).

[212] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. "Chain of Thought Prompting Elicits Reasoning in Large Language Models". In: *NeurIPS*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho. 2022 (cit. on pp. 3, 25).

[213] Jason Wei and Kai Zou. "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks". In: *EMNLP-IJCNLP*. 2019 (cit. on p. 12).

[214] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big data* 3 (2016), pp. 1–40 (cit. on p. 2).

[215] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. "Robust fine-tuning of zero-shot models". In: *CVPR*. 2022, pp. 7959–7971 (cit. on p. 39).

[216] Junyang Wu, Xianhang Li, Chen Wei, Huiyu Wang, Alan Yuille, Yuyin Zhou, and Cihang Xie. "Unleashing the Power of Visual Prompting At the Pixel Level". In: *arXiv:2212.10556* (2022) (cit. on p. 12).

[217] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. "Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly". In: *IEEE transactions on pattern analysis and machine intelligence* 41.9 (2018), pp. 2251–2265 (cit. on pp. 2, 66, 72).

[218] Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. "SUN Database: Exploring a Large Collection of Scene Categories". In: *IJCV* (2016) (cit. on pp. 42, 59, 118).

[219] Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, and Yanning Zhang. "Dual modality prompt tuning for vision-language pre-trained model". In: *arXiv:2208.08340* (2022) (cit. on p. 12).

[220] Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. "Energy-Based Concept Bottleneck Models". In: *The Twelfth International Conference on Learning Representations*. 2023 (cit. on p. 67).

[221] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. "Language in a bottle: Language model guided concept bottlenecks for interpretable image classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 19187–19197 (cit. on p. 66).

[222] Teresa Yeo, Andrei Atanov, Harold Benoit, Aleksandr Alekseev, Ruchira Ray, Pooya Esmaeil Akhoondi, and Amir Zamir. "Controlled training data generation with diffusion models". In: *arXiv preprint arXiv:2403.15309* (2024) (cit. on p. 52).

[223] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems* 27 (2014) (cit. on p. 2).

[224] Zebin You, Yong Zhong, Fan Bao, Jiacheng Sun, Chongxuan Li, and Jun Zhu. "Diffusion models and semi-supervised learners benefit mutually with few labels". In: *NeurIPS* (2023) (cit. on p. 52).

[225] Aron Yu and Kristen Grauman. "Fine-grained visual comparisons with local learning". In: *CVPR*. 2014 (cit. on pp. 28, 112).

[226] Aron Yu and Kristen Grauman. "Semantic jitter: Dense supervision for visual comparisons via synthetic images". In: *ICCV*. 2017 (cit. on pp. 28, 112).

[227] Zhuoran Yu, Chenchen Zhu, Sean Culatana, Raghuraman Krishnamoorthi, Fanyi Xiao, and Yong Jae Lee. "Diversify, Don't Fine-Tune: Scaling Up Visual Recognition Training with Synthetic Images". In: *arXiv preprint arXiv:2312.02253* (2023) (cit. on pp. 36, 50, 52, 55).

[228] Jianhao Yuan, Jie Zhang, Shuyang Sun, Philip Torr, and Bo Zhao. "Real-Fake: Effective Training Data Synthesis Through Distribution Matching". In: *ICLR*. 2024 (cit. on p. 52).

[229] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. "When and Why Vision-Language Models Behave like Bags-Of-Words, and What to Do About It?" In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=KRLUvxh8uaX (cit. on pp. 3, 17).

[230] Mert Yuksekgonul, Maggie Wang, and James Zou. "Post-hoc Concept Bottleneck Models". In: *The Eleventh International Conference on Learning Representations*. 2022 (cit. on p. 66).

[231] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. "Cutmix: Regularization strategy to train strong classifiers with localizable features". In: *iccv*. 2019 (cit. on pp. 2, 116).

[232] Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, et al. "Concept embedding models". In: *arXiv preprint arXiv:2209.09056* (2022) (cit. on pp. 64–68, 70, 72, 75).

[233] Mateo Espinosa Zarlenga, Pietro Barbiero, Zohreh Shams, Dmitry Kazhdan, Umang Bhatt, Adrian Weller, and Mateja Jamnik. "Towards robust metrics for concept representation evaluation". In: *arXiv preprint arXiv:2301.10367* (2023) (cit. on pp. 67, 68, 70).

[234] Mateo Espinosa Zarlenga, Katherine M Collins, Krishnamurthy Dj Dvijotham, Adrian Weller, Zohreh Shams, and Mateja Jamnik. "Learning to Receive Help: Intervention-Aware Concept Embedding Models". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023 (cit. on pp. 64–68, 70–72, 75).

[235] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. "mixup: Beyond Empirical Risk Minimization". In: *ICLR*. 2018 (cit. on pp. 12, 116).

[236] Kaiwen Zhang, Yifan Zhou, Xudong Xu, Bo Dai, and Xingang Pan. "DiffMorpher: Unleashing the Capability of Diffusion Models for Image Morphing". In: *CVPR*. 2024 (cit. on p. 52).

[237] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. "Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners". In: *CVPR*. 2023 (cit. on pp. 3, 52).

[238] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. "Tip-adapter: Training-free adaption of clip for few-shot classification". In: *ECCV*. 2022 (cit. on pp. 3, 57, 59).

[239] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. "A survey of large language models". In: *arXiv preprint arXiv:2303.18223* (2023) (cit. on p. 24).

[240] Chenyu Zheng, Guoqiang Wu, and Chongxuan Li. "Toward understanding generative data augmentation". In: *NeurIPS* (2023) (cit. on p. 52).

[241] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. "Places: A 10 million Image Database for Scene Recognition". In: *IEEE TPAMI* (2017) (cit. on p. 16).

[242] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. "Conditional prompt learning for vision-language models". In: *CVPR*. 2022 (cit. on p. 12).

[243] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. "Learning to prompt for vision-language models". In: *IJCV* (2022) (cit. on pp. 3, 10, 12, 24, 29, 41, 43–45, 57, 59, 112).

[244] Yongchao Zhou, Hshmat Sahak, and Jimmy Ba. "Training on thin air: Improve image classification with generated data". In: *arXiv preprint arXiv:2305.15316* (2023) (cit. on pp. 36, 38, 52, 53).

[245] Xiaojin Jerry Zhu. "Semi-supervised learning literature survey". In: (2005) (cit. on p. 2).

[246] Roland S. Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. "Contrastive Learning Inverts the Data Generating Process". In: *ICML*. 2021 (cit. on p. 21).

A

Zero-shot Learning by Leveraging Foundation Models

A.1 Waffling around for Performance: Visual Classification with Random Words and Broad Concepts

In this supplementary material, we first provide a collection of additional results in §A.1.1 which extend those presented in the main paper to more backbone models. Finally, we showcase the GPT-generated descriptors for our additionally used benchmarks beyond [128] (§A.1.2), and present some exemplary images from the eleven benchmarks used in this work in Fig. A.1.

A.1.1 Additional results

Motivational experiments for random class descriptors. In Tab. A.1, we extend our motivational experiments on random class descriptor assignment to motivate `WaffleCLIP` from Tab. 2.1, highlighting similar behaviour on both a larger ViT-L/14 and a ResNet50 backbone network. Descriptor randomization does not result in a significant drop in performance, but rather yields performances that match DCLIP.

Comparison of `WaffleCLIP` and DCLIP. Tab. A.2 extends results from Tab. 2.2 on the ViT-L/14 and ResNet50 backbones, in which `WaffleCLIP` as a standalone method, as well as equipped with high-level concepts and/or joint usage of LLM-generated descriptors, is compared to DCLIP. The results confirm our conclusions drawn in §2.1.4.1, wherein `WaffleCLIP`, without access to any external LLM, can match the performance of LLM-descriptor-based approaches like DCLIP. In addition to that, we again find complementarity of randomized descriptors and LLM-generated descriptors. Furthermore, we observe performance gains through the usage of automatically generated high-level concepts.

ViT-L/14	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	**Avg**
CLIP [155]	67.90	73.37	62.24	56.03	40.46	92.55	93.30	52.87	67.34
DCLIP [128]	**69.72**	75.26	63.53	58.72	**42.60**	92.81	93.89	**56.60**	69.14
DCLIP (same, 1x)	69.27 ±0.23	75.05 ±0.15	64.21 ±0.36	57.59 ±1.72	42.01 ±0.23	93.15 ±0.13	93.97 ±0.22	55.16 ±0.47	68.80 ±0.66
DCLIP (same, 2x)	69.58 ±0.21	**75.30** ±0.16	**64.30** ±0.26	**59.32** ±1.63	42.28 ±0.17	**93.31** ±0.05	**94.04** ±0.11	55.31 ±0.50	**69.18** ±0.62
ResNet50	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	**Avg**
CLIP [155]	51.34	58.16	45.20	28.09	36.63	78.37	83.76	38.51	52.51
DCLIP [128]	52.70	59.66	**47.76**	34.27	38.39	78.59	**85.77**	**41.01**	54.77
DCLIP (same, 1x)	52.63 ±0.28	59.69 ±0.30	**47.76** ±0.39	32.74 ±1.49	38.63 ±0.22	80.08 ±0.58	85.36 ±0.52	40.77 ±0.63	54.71 ±0.67
DCLIP (same, 1x)	**52.89** ±0.23	**59.90** ±0.26	47.70 ±0.29	**34.37** ±1.27	**38.93** ±0.21	**80.11** ±0.30	85.34 ±0.29	40.91 ±0.79	**55.02** ±0.58

Table A.1: **Motivating random class descriptors - additional backbones.** Extension of our motivational experiments from Tab. 2.1 with ViT-L/14 and ResNet50 backbones.

ViT-L/14	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	**Avg**
CLIP [155]	67.90	73.37	62.24	56.03	40.46	92.55	93.30	52.87	67.34
+ Concepts	↓	↓	63.01	61.23	41.07	93.52	93.65	↓	68.32
DCLIP [128]	**69.72**	75.26	63.53	58.72	42.60	92.81	93.89	**56.60**	69.14
WaffleCLIP (ours)	69.48 ±0.08	75.30 ±0.04	**64.18** ±0.13	**61.17** ±0.35	42.26 ±0.10	93.31 ±0.09	91.98 ±0.11	53.94 ±0.29	68.95 ±0.18
+ Concepts	↓	↓	63.40 ±0.17	60.20 ±0.87	42.57 ±0.09	**93.65** ±0.05	**94.38** ±0.08	↓	69.12 ±0.33
+ GPT descr.	**69.80** ±0.13	**75.57** ±0.06	**64.32** ±0.21	**60.63** ±1.23	**42.96** ±0.12	93.28 ±0.08	93.35 ±0.22	**56.33** ±0.42	**69.53** ±0.48
+ GPT descr. + Concepts	↓	↓	63.14 ±0.16	**61.82** ±1.07	**42.95** ±0.09	93.49 ±0.04	94.12 ±0.09	↓	**69.65** ±0.42
ResNet50	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	**Avg**
CLIP [155]	51.34	58.16	45.20	28.09	36.63	78.37	83.76	38.51	52.51
+ Concepts	↓	↓	46.60	34.06	37.43	80.89	83.43	↓	53.80
DCLIP [128]	**52.70**	59.66	47.76	34.27	38.39	78.59	**85.77**	**41.01**	54.77
WaffleCLIP (ours)	**52.89** ±0.15	**60.12** ±0.12	47.68 ±0.15	31.34 ±0.47	38.32 ±0.10	79.68 ±0.17	84.32 ±0.20	39.25 ±0.27	54.20 ±0.23
+ Concepts	↓	↓	**48.34** ±0.13	35.08 ±0.42	39.03 ±0.08	**81.38** ±0.08	**85.80** ±0.12	↓	55.24 ±0.21
+ GPT descr. + Concepts	↓	↓	**48.41** ±0.21	**37.36** ±0.62	**39.43** ±0.07	81.17 ±0.09	**85.82** ±0.16	↓	**55.75** ±0.26

Table A.2: **Performance of WaffleCLIP with additional backbones.** Here, we extend the comparison of WaffleCLIP (Tab. 2.2) to GPT-generated fine-grained class descriptors in DCLIP [128] for ViT-L/14 and ResNet50 backbones. We find similarly consistent insights, where our LLM-free WaffleCLIP can match the performance of DCLIP. Joint usage of both randomized and LLM-generated descriptors again reveals complementarity (WaffleCLIP + *GPT descr*). In addition to that, the usage of automatically extracted high-level semantic concepts can provide consistent additional performance gains (+ *Concepts*). We use (↓) to denote the same results as previous lines where high-level concept guidance is not applicable.

Progression from systematic to fully randomized descriptor scrambling. Tab. A.3 extends the descriptor scrambling progression studies from Tab. 2.4 to two additional backbones, namely, ViT-L/14 and ResNet50. Similar to the ViT-B/32 backbone, a move from systematic semantic shifts to independently subsampled descriptors can recover and even beat the performance of DCLIP.

A.1.2 Exemplary GPT-3 generated descriptors for additional benchmarks

As we introduce descriptions for three additional datasets beyond those used in [128], we provide four example descriptors for three random classes in each dataset.

Flowers102

Pink Primrose

- `"delicate flower"`
- `"five petals in a star shape"`

ViT-L/14	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	**Avg**
DCLIP [128]	**69.72**	**75.26**	63.53	**58.72**	**42.60**	92.81	93.89	**56.60**	**69.14**
DCLIP (interchanged)	66.44 ±0.12	72.07 ±0.15	63.62 ±0.44	51.49 ±4.89	37.06 ±0.41	91.30 ±0.30	93.74 ±0.28	49.84 ±0.78	65.69 ±1.77
DCLIP (scrambled)	68.68 ±0.21	74.47 ±0.11	63.78 ±0.13	55.98 ±2.01	41.29 ±0.23	92.29 ±0.20	93.52 ±0.18	53.28 ±1.12	67.91 ±0.83
DCLIP (random, 1x)	68.01 ±0.22	73.89 ±0.08	63.81 ±0.22	55.72 ±2.01	40.32 ±0.29	92.37 ±0.31	93.60 ±0.19	52.83 ±0.46	67.57 ±0.76
DCLIP (random, 5x)	69.27 ±0.17	75.11 ±0.08	64.25 ±0.16	58.34 ±1.55	42.11 ±0.14	93.22 ±0.12	93.88 ±0.09	55.28 ±0.23	68.93 ±0.57
ResNet50	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	**Avg**
DCLIP [128]	52.70	59.66	**47.76**	34.27	38.39	78.59	**85.77**	**41.01**	54.77
DCLIP (interchanged)	49.80 ±0.22	56.35 ±0.06	47.68 ±0.32	28.17 ±4.43	33.77 ±0.34	77.59 ±0.29	84.60 ±0.63	35.81 ±1.12	51.72 ±1.64
DCLIP (scrambled)	52.20 ±0.20	59.21 ±0.06	47.60 ±0.39	34.98 ±2.00	37.90 ±0.18	78.33 ±0.14	85.07 ±0.34	39.19 ±0.95	54.31 ±0.81
DCLIP (random, 1x)	51.60 ±0.29	58.29 ±0.15	47.37 ±0.23	30.18 ±4.18	36.82 ±0.26	78.87 ±0.24	84.52 ±0.17	38.89 ±0.85	53.32 ±1.52
DCLIP (random, 5x)	52.81 ±0.09	59.73 ±0.05	47.74 ±0.10	34.53 ±0.74	38.62 ±0.15	80.20 ±0.13	85.30 ±0.15	40.29 ±0.46	54.90 ±0.32

Table A.3: **Progression from systematic to fully randomized descriptor scrambling - additional backbones.** We extend our descriptor scrambling progression studies from Tab. 2.4 to two additional backbones: ViT-L/14 and ResNet50. In both cases, the same trend can be seen, in which a move from systematic semantic shift to independently subsampled descriptors can recover the performance of DCLIP after an initial performance drop.

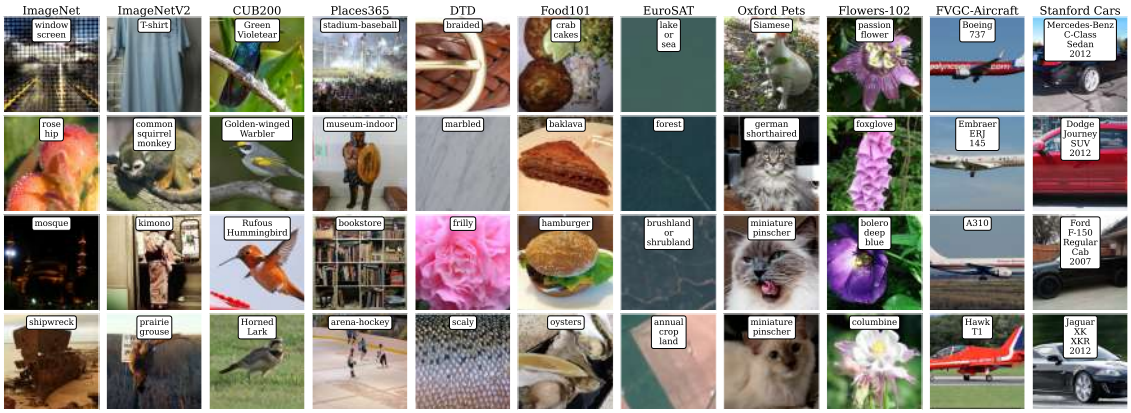


Figure A.1: To get an intuition of the different visual classification tasks, we showcase samples of four randomly selected classes for each of the eleven utilized visual classification benchmarks.

- `"pink in color"`
- `"often has yellow center"`

Balloon Flower

- `"a delicate flower with five petals"`
- `"a unique balloon-like shape"`
- `"a star-shaped center in the middle of the flower"`
- `"vibrant colors such as pink, purple, blue, white, and yellow"`

Sunflower

- `"large, bright yellow petals"`
- `"a dark center surrounded by disk florets"`

- "long stem"
- "a single, long, narrow leaves tapered to a point"

FGVCAircraft

A300

- "black or silver color"
- "a rectangular body with rounded edges"
- "two lens ports"
- "a mode dial"

EMB-120

- "a cabin with 30-33 seats"
- "a distinctive high-wing design"
- "two Pratt and Whitney PW118 turboprop engines"
- "a T-tail configuration"

Tornado

- "dark, rotating funnel-shaped cloud"
- "strong winds"
- "dark clouds"
- "heavy precipitation"

Stanford Cars

Acura TL Sedan 2012

- "silver, grey, or black exterior"
- "Acura logo on the front grille"
- "distinctive headlights"
- "chrome accents on the exterior"

BMW X6 SUV 2012

- "four-door SUV"

- "sloping roof-line"
- "signature BMW kidney grille"
- "round headlights and taillights"

Honda Odyssey Minivan 2012

- "four doors and a hatchback"
- "a curved hood"
- "wide, round headlights"
- "a Honda logo"

A.2 Feasibility with Language Models for Open-World Compositional Zero-Shot Learning

A.2.1 Broader Impact and Limitations

Determining the feasibility of state-object pairs in all combinations is crucial when deploying the model. By accurately assessing feasibility, we prevent the model from predicting unrealistic classes, thus improving the model performance and reducing negative impacts on end-users. However, our FLM is limited in that we use prior knowledge, i.e. seen classes from the training dataset, which could be biased. If the seen classes represent only part of the semantics, e.g. only the texture-related attributes like "furry" or "zigzag" for animal in the seen classes while age-related states exist in the test set, our method may struggle to predict the true feasible pairs accurately. The bias can also lead to fairness issues if the seen classes only represent certain groups. Therefore, it is important to curate an unbiased seen class set to support the model's understanding of feasibility for the query pairs.

A.2.2 Prompt search in FLM

There are four prompt components in FLM: persona, instruction, guidance, and query. In our experiments, We keep the *persona* component fixed as "`You are a helpful, respectful and honest assistant.`". We conduct a grid search using four instruction, four guidance, and four query sentences, which are:

instruction_list = {
- "`Answer with a single word, yes or no.`",
- "`Answer with a single word, yes or no, followed by an explanation.`",
- "`Answer with yes or no.`",
- "`Answer with yes or no, followed by an explanation.`",

},

guidance_list = {
- "`The following list consists of words that fit together.`",
- "`The following list consists of word combinations that make sense.`",
- "`The given list consists of word combinations that make sense.`",
- "`The given list comprises word combinations that make sense.`",

},

query_list = {
- "`Considering the list above, does "{`*s*`} {`*o*`}" fit into the list?`",
- "`Does "{`*s*`} {`*o*`}" fit into the list above?`",
- "`Does "{`*s*`} {`*o*`}" align with the contents of the list provided above?`",
- "`Considering the list above, does "{`*s*`} {`*o*`}" align with the contents?`",

},

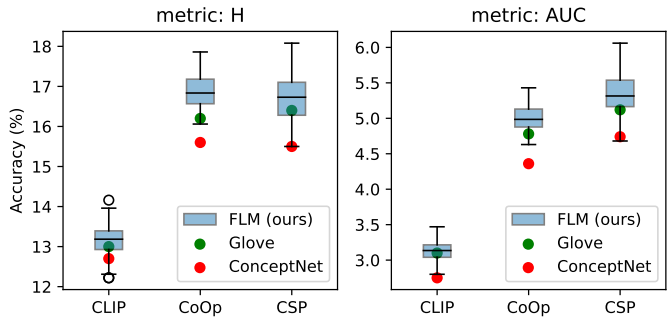


Figure A.2: Prompt variation results on MIT-States dataset.

and select the combination that yields the highest unseen validation accuracy following the validation protocol and split of [135]. On MIT-States, we use "`Answer with a single word, yes or no, followed by an explanation.`" as instruction, "`The following list consists of words that fit together.`" as guidance, and "`Does "{s} {o}" fit into the list above?`" as query. On the UT-Zappos, we use "`Answer with a single word, yes or no.`" as instruction, "`The given list consists of word combinations that make sense.`" as guidance, and "`Considering the list above, does "{s} {o}" fit into the list?`" as query. Finally on the C-GQA, we use "`Answer with a single word, yes or no, followed by an explanation.`" as instruction, "`The given list consists of word combinations that make sense.`" for CLIP and "`The given list comprises word combinations that make sense.`" for CoOp and CSP as guidance, and "`Does "{s} {o}" align with the contents of the list provided above?`" as query.

To examine a broader range of prompt variations, we report the results on MIT-States as a box plot in Figure A.2. We first observe the results vary with different prompts. For instance, the Harmonic mean accuracy in CSP ranges from 15.5% to 18.1%. However, despite this variability, most of the prompts outperform the baselines. For instance, the Glove result always lies close to or below the lower quartile (25th percentile) of the box plot, and ConceptNet even further below that.

A.2.3 Benchmarks and Hyperparameter settings

Benchmarks. We use three standard datasets for OW-CZSL, i.e., MIT-States [82], UT-Zappos [225, 226], and C-GQA [132]. Each dataset comprises a set of states and objects, where an object-state combination forms a class. MIT-States consists of 115 states and 245 objects, resulting in a total of 28,175 possible pairs. Among these possible pairs, 1,262 and 700 pairs are seen classes and unseen classes, respectively. UT-Zappos includes 16 states and 12 objects, leading to 192 possible pairs, with 83 seen classes and 33 unseen classes. Finally, C-GQA has 413 states and 674 objects, resulting in 278,362 possible pairs, with 5,592 seen classes and 1,963 unseen classes.

Hyperparameter settings. We train CoOp [243] and CSP [135] with the CSP official code [1]

[1] https://github.com/BatsResearch/csp

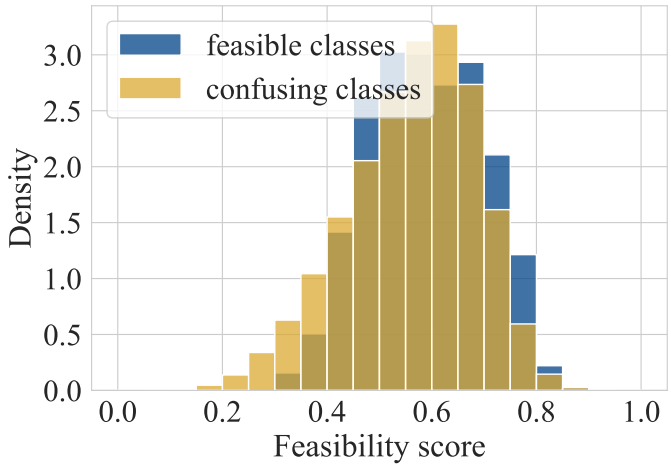


Figure A.3: Feasibility scores for question-answer "yes" format in the in-context prompt.

to get the fine-tuned models. Following the original CSP setting, we fine-tune a pretrained CLIP [155] model with a ViT-L/14 backbone for 20 epochs and choose the checkpoint with the highest validation accuracy on unseen classes. During training, we employ a batch size of 64, a learning rate of 5e-4, and a weight decay of 1e-5. Additionally, we set the attribute dropout rate for CSP to 0.3. We did not optimize these hyperparameters as originally done by [135]. The model training is performed on a single A100 GPU. Similarly, when querying Vicuna-13B for our FLM, we use a single A100 GPU.

A.2.4 Question-Answer format: 0-9 score

As mentioned in the main text, we observe that the prompt ablation "Format QA: yes" drops performance across all datasets, e.g. 17.4%→12.0 and 6.55%→3.69% in harmonic mean on MIT-States and C-GQA. The lower performance originates from the tendency of the LLM to answer "Yes" since the provided examples in the guidance are always positive. This phenomenon is also evident in Figure A.3, where both the distributions of feasible and confusing classes overlap and lean closer towards 1. These observations indicate that employing a list format for guidance rather than a question-answer format is crucial in obtaining accurate feasibility scores.

MIT-States	Method			C-GQA	Method		
class	GloVe	ConceptNet	FLM (ours)	class	GloVe	ConceptNet	FLM (ours)
folded book	✘	✔	✔	blue table	✔	✘	✔
rusty truck	✘	✔	✔	brown cake	✔	✔	✘
small dog	✘	✘	✔	balding person	✘	✔	✔
eroded granite	✔	✔	✘	blue tray	✘	✘	✔
gray stove	✔	✘	✔	asian person	✘	✔	✔
thick ring	✘	✘	✘	yellow leaf	✘	✔	✔

Table A.4: Qualitative examples from the MIT-States and C-GQA datasets. A state-object pair is deemed feasible (✔) or infeasible (✘) by the respective methods.

Similar trends are observed for "Format QA: score" where we use a question-answer format with guiding LLMs to respond with an integer score instead of a binary answer. To obtain an integer score as an answer from the LLMs, we construct the guidance component of the human message as:

"The following list consists of words and their likelihood of existence in the real world, scored on a scale of 0 to 9.
- {s_1} {o_1}, score: 9
- {s_2} {o_2}, score: 9
- ...
- {s_n} {o_n}, score: 9

What is the score for "{s} {o}"?"

If we had access to a more nuanced classification of prior feasibility scores for the seen classes, we could provide the LLM with a more informative guidance. As this is not available, we had to choose fixed score value. The results are shown in the "Format QA: score" row of Table 2.11 in the main text. We observe the performance drops across the datasets, e.g. 17.4%→12.3% and 6.55%→3.69% in harmonic mean on MIT-States and C-GQA. These observations, together with "QA: yes" format, indicate that employing a list format for guidance rather than a question-answer format is crucial in obtaining accurate feasibility scores.

A.2.5 Qualitative examples.

We examine qualitative examples of feasibility classifications where we compare the predictions of three methods in Table A.4. It displays unseen classes most commonly found in the test datasets. We observe that in both the MIT-States and C-GQA datasets, our FLM and ConceptNet tend to predict the given classes as feasible more frequently compared to GloVe. One interesting observation arises from the class "small dog" which is surprisingly predicted as infeasible in both GloVe and ConceptNet, but correctly identified as feasible by FLM.

B

DataDream: Few-shot Guided Dataset Generation

A Broader Impacts and Limitations

While DataDream primarily focuses on generating synthetic image datasets for image classification, our approach can be extended to other domains and tasks. For example, sentiment classification can benefit from synthetic datasets generated by large language models fine-tuned on a few challenging samples. Additionally, DataDream can be applied to other image modalities. For instance, the scarcity of medical data often impedes the performance of medical AI. Our method could serve as an augmentation tool to enhance performance in such scenarios.

However, it's important to recognize the limitations of our approach. We observe, in EuroSAT and DTD, a performance gap when comparing models trained solely on DataDream synthetic datasets to those trained solely on real few-shot data, with the latter performing better. We speculate that this disparity comes from the challenges generative models face when learning from out-of-distribution data. It becomes difficult to fine-tune generative models with few-shot samples that are different from the data the models were originally trained on. For instance, satellite land images, those in the EuroSAT dataset, may not be well-represented in the LAION dataset used for training Stable Diffusion models. Consequently, the diffusion model struggles to generalize and accurately interpret satellite images, even after fine-tuning with real few-shot satellite images.

Moreover, by training on synthetic training, our method inherits the limitations of the underlying generative models. For instance, generative models will propagate biases that appear in the training data, such as social or gender biases. As a result, classifiers trained on the synthetic data of a generative models are also prone to carrying forward these biases. Especially when employing proprietary generative models, it is often unknown what data they were trained on.

DataDream fine-tunes the generative model using few-shot samples. Because the dataset is much smaller, it is easier to control for biases that could potentially be introduced

in this stage, but this would require manual intervention when curating the few-shot dataset. At the same time, malicious actors could purposefully introduce biases through the DataDream fine-tuning, for example to construct a classifier that discriminates against minorities. Detecting biases in the resulting classifier might then become more difficult than observing them from generative images. We believe that investigating and mitigating bias in synthetic data generation would be an important area for future research.

B Implementation details

Following the methodology of DISEF [37] for the classifier training, we consider different learning rates, weight decay, and whether to use Mixup [235] and Cutmix [231] methods for data augmentation as a hyper-parameter. Concretely, we use batch size 64, and AdamW [118] as an optimizer with a cosine annealing scheduler. The learning rate is searched in $\{1e-4, 1e-5, 1e-6, 1e-7\}$ and the weight decay in $\{5e-4, 1e-4\}$. We use standard augmentation methods as default, i.e. random resized crop, random horizontal flip, random color jitter, and random gray scale, while searching on whether to additionally use Mixup and Cutmix. We set the weight assigned to the real loss term as $\lambda = 0.8$.

C Baseline methods

We compare our DataDream method to two state-of-the-art image generation methods in the few-shot setting, IsSynth [69] and DISEF [37]. While these methods originally use GLIDE [137] and Stable Diffusion v1.5 [159], respectively, for the generative model, we use the Stable Diffusion v2.1 and follow the other pipelines suggested in each paper. We implement both methods based on DISEF official code[1]. When running the IsSynth method, we replace the textual prompt conditioned on the diffusion model from the caption generated by LLaVA [110] to a standard prompt, e.g. `"a photo of a [CLS]"`, as suggested in IsSynth. When we compare DataDream to IsSynth and DISEF in Table 3.1, we use the same generation procedure of generating 500 images without filtering out them but using them all for training the classifier.

D Compatibility with other CLIP fine-tuning and classifiers

DataDream generates data for downstream tasks, so it is compatible with other CLIP fine-tuning methods. We evaluate the synthetic dataset on the most recent suggested method for training the CLIP model [211]. Moreover, we evaluate our and other datasets with two additional classifiers: CLIP ViT-L/14 and CLIP RN50. For both, we fully fine-tune the visual encoder and LoRA fine-tune the text encoder. As shown in the table B, DataDream works better than the baselines in both CLIP fine-tuning method [211] and other classifiers for both sythetic only and real + synthetic settings.

[1] https://github.com/vturrisi/disef

	R	S	[A] AirC	Cars	Food	CAL	CLIP ViT-L/14 AirC	Cars	Food	CAL	CLIP RN50 AirC	Cars	Food	CAL
IsSynth [13]	✓		24.09	69.59	84.79	94.58	30.52	81.64	90.18	97.01	27.22	74.42	48.78	87.33
DISEF [9]	✓		26.03	63.82	84.86	93.90	31.59	71.55	90.20	96.40	22.25	33.36	46.03	83.70
DataDream$_{cls}$	✓		39.97	79.90	85.44	**94.63**	69.21	94.21	**91.72**	97.78	67.79	92.31	61.03	90.05
DataDream$_{dset}$	✓		**43.75**	**81.37**	**85.20**	94.62	**76.33**	**94.53**	91.51	**97.94**	**75.31**	**93.34**	**64.16**	**91.22**
base (fewshot)	✓		40.61	75.12	79.05	92.55	72.01	91.18	91.92	**98.46**	61.57	78.86	63.52	93.29
IsSynth [13]	✓	✓	43.10	80.50	86.30	95.48	73.33	93.68	91.96	98.1	70.94	90.82	68.77	94.54
DISEF [9]	✓	✓	42.62	79.41	85.97	95.40	73.83	92.67	91.94	98.30	65.99	79.18	**70.10**	94.34
DataDream$_{cls}$	✓	✓	48.41	83.78	86.67	95.47	77.69	94.71	**92.16**	98.22	79.21	92.99	66.70	94.37
DataDream$_{dset}$	✓	✓	**49.31**	**83.87**	**86.71**	**95.62**	**80.98**	**95.18**	92.14	98.30	**81.46**	**93.30**	66.63	**94.62**

Table B: Compatibility with other CLIP fine-tuning and classifiers.

E Ablation study

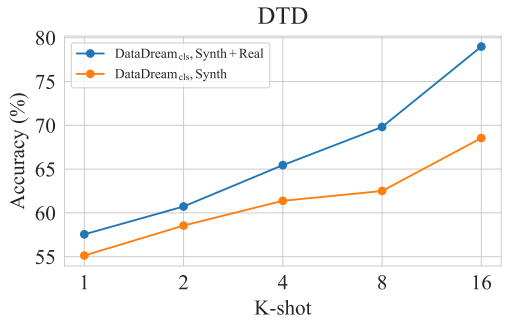
Method	txt enc.	w/o pre. loss	Synth	Real+Synth
DataDream$_{dset}$			19.96	62.49
DataDream$_{dset}$	✓		18.58	62.61
DataDream$_{dset}$		✓	59.80	66.78
DataDream$_{dset}$ (ours)	✓	✓	**71.07**	**71.98**

Table B.2: **Ablation study of using different pipeline for DataDream training.** Mark on "txt enc." indicates training LoRA on the text encoder in addition to the UNet (no mark = UNet only). Mark on "w/o pre. loss" indicates using preservation loss [168] in addition to the reconstruction loss (Equation 3.3).

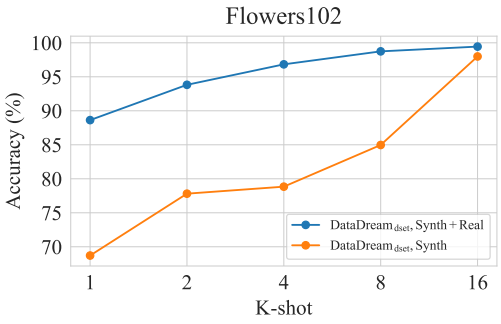
For our DataDream method, we take inspiration from DreamBooth [168] which proposes to finetune a diffusion model to generate personalized images. To enable the generation of personalized images across diverse prompts beyond the one it was trained on, e.g. `"a photo of a [CLS]"`, DreamBooth introduces a preservation loss. This loss acts as a regularizer such that the fine-tuned model maintains its original capabilities when generating images in the absence of personalized tokens in the textual prompt. Since we are not interested in employing our fine-tuned model for general-purpose image generation, we put more focus on faithful replication of the few-shot data distribution than preserving the generation quality of irrelevant generations.

We conduct an ablation study to investigate the impact of preservation loss on data generation within DataDream. Additionally, We explore the effect of applying LoRA solely to the UNet, or to both the UNet and text encoder of the diffusion model. We train each combination setting for 200 epochs on the FGVC Aircraft [121] dataset, and the results are shown in Table B.2.

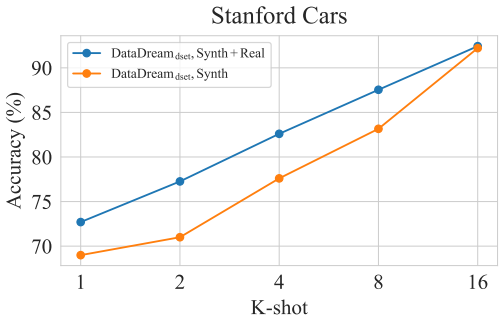
First, we observe that applying LoRA to both the UNet and text encoder (without preservation loss) improves the accuracy from 59.80% to 71.07% on the classifier trained solely with synthetic data, and 66.78% to 71.98% when incorporating real 16-shot data with synthetic data. This improvement indicates that additional fine-tuning of the text encoder enhances the model's ability to capture the features present in few-shot data. Second, we observe that using the preservation loss decreases the accuracy from 71.07% to 18.58% in the synthetic setting and 71.98% to 62.61% in the real+synth setting. This decline suggests that, the inclusion of preservation loss hinders the model's adaptation to the target few-shot data, limiting its generation performance. Furthermore, given that we employ the same standard prompt for both training DataDream and generating images with the DataDream-fintuned model, the necessity for the preservation loss for



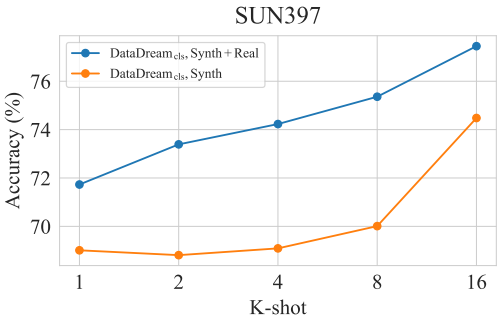
(a) **DataDream$_{cls}$ accuracy scaling by number of shots for DTD** [36].



(b) **DataDream$_{dset}$ accuracy scaling by number of shots for Flowers102** [139].



(c) **DataDream$_{dset}$ accuracy scaling by number of shots for Stanford Cars** [100].



(d) **DataDream$_{cls}$ accuracy scaling by number of shots for Sun397** [218].

DataDream training is diminished. Overall, our findings suggest that using LoRA adaptor in the text encoder and excluding the preservation loss achieve the best performance.

F *K*-shot varying *K* on additional datasets

In this section, we include additional scaling graphs on four additional datasets, which show DataDream performance improves as we increase K in the K-shot setting. The included datasets are DTD [36] (Figure B.1a), Flowers102 [139] (Figure B.1b), Stanford Cars [100] (Figure B.1c), and SUN397 [218] (Figure B.1d). For each dataset, we show results with the better of our two models, as reported in Table 3.1: DataDream$_{dset}$ for Flowers102 and Stanford Cars, and DataDream$_{cls}$ for DTD [36] and SUN397 [218]. We include results for both using only synthetic (Synth) as well as the combination of synthetic and real data (Synth+Real).

One aspect we would like to highlight is the performance similarity on Stanford Cars [100] between the Synth and Real+Synth settings on 16 shots. This can already be seen in Table 3.1, but especially stands out in Figure B.1c. This means that by sixteen shots, DataDream$_{dset}$ can faithfully represent the information from the real data, to the point where there is no performance gained from additionally training on the real samples.

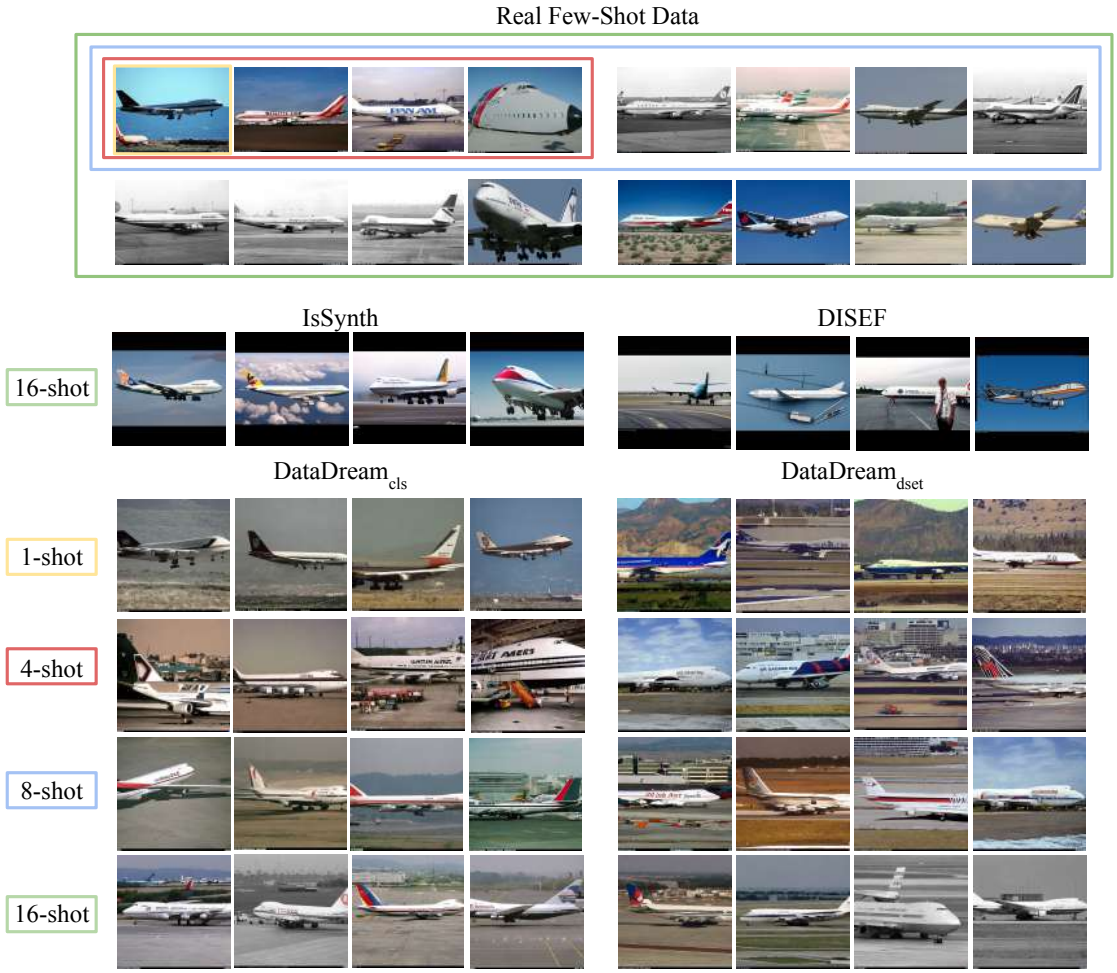


Figure B.2: **Qualitative results of the class 747-100 from the FGVC Aircraft** [121] dataset, created the same as Figure 3.3.

G Qualitative examples

We provide additional qualitative results for one further FGVC Aircraft class and two extra datasets. As done in Figure 3.3, we designate the real few-shot images provided for 1-, 4-, 8-, and 16-shots. We then show the 16-shot images generated by two competing methods, IsSynth [69] and DISEF [37]. Finally, we have images generated by both DataDream$_{dset}$ and DataDream$_{cls}$, for varying number of shots.

In Figure B.2, we see images from the class 747-100 in FGVC Aircraft [121], which is a commercial aircraft variety. When comparing with previous SOTA, we once again see that DataDream generally provides better in-distribution data. DISEF in particular generates many incorrect modalities (e.g. cartoon or toy) or images with irrelevant primary subjects. In comparison, DataDream consistently generates images with the correct shape, as seen from angles commonly found in the dataset.

In Figure B.3, we show images of Stanford Cars' [100] Volkswagen Beetle Hatchback

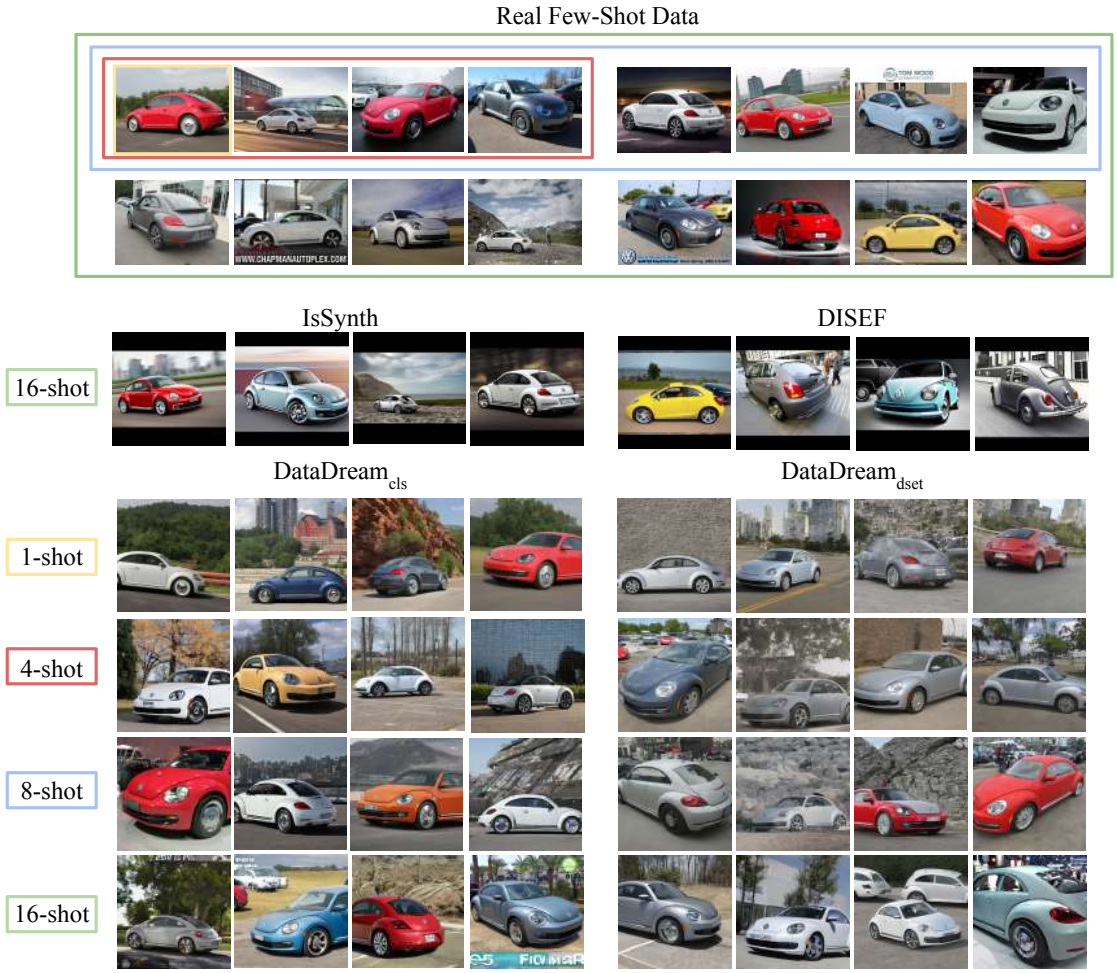


Figure B.3: **Qualitative results of the class Volkswagen Beetle HatchBack 2012 from the Stanford Cars** [100] dataset, created the same as Figure 3.3.

2012 class. Once again, we see that DataDream is better at consistently generating the correct car than DISEF, and generates backgrounds closer to what is found in the real dataset than IsSynth. Furthermore, we would like to point out a coloring difference between DataDream$_{dset}$ and DataDream$_{cls}$. We know that Volkswagen Beetles are available in a wide variety of colors; DataDream$_{cls}$ demonstrates this by generating cars in yellow, orange, multiple shades of bright blue, etc. On the other hand, most car varieties are available in a smaller color pool, many of which are muted. We see that DataDream$_{dset}$-generated cars are more likely to be white, gray, or red, all of which are colors commonly found in other cars. There are still several shades of blue generated, but they are more muted than those generated by DataDream$_{cls}$. Hence, we see an example of how DataDream$_{dset}$ can learn patterns from the wider dataset and apply them to individual classes where it may not be optimal, while DataDream$_{cls}$ can differentiate better. We remember from Table 3.1 that DataDream$_{dset}$ performed better than DataDream$_{cls}$ on the full dataset, so overall, the value of information shared between classes was greater than what was lost.

Figure B.4: **Qualitative results of the class Sword Lily from the Flowers102** [139] dataset, created the same as Figure 3.3.

Finally, we provide examples of generated images from the Sword Lily class in the Flowers102 dataset [139]. First, we notice that while both previous methods struggle to generate faithful representations of the class, DataDream$_{cls}$ generates accurate images from a single shot, and DataDream$_{dset}$ from four shots. One interesting aspect is the number of flowers per image. At a single and four shots, DataDream$_{cls}$ generates a few images of single or double flowers. However, by four-shots for DataDream$_{cls}$ and from the first shot for DataDream$_{dset}$, flowers are almost always generated in bunches, outside. While this is representative of the majority of images, there are also examples in the real dataset with a low number of flowers. Hence, there may still be more to gain in terms of ensuring that the entire distribution is represented in the synthetic dataset. We leave this for future work to explore.

C

LoFT: LoRA-fused Dataset Generation with Few-shot Guidance

A Implementation details of classifier training

When fine-tuning the CLIP model, we freeze the CLIP parameters and update the LoRA weights applied to them, with a lora rank of 16. We use a batch size of 256 and a learning rate of 1e-6 with cosine annealing for the learning rate schedule. The weight decay is set to 1e-4. As the dataset size increases, we adjust the number of iterations to be increased while decreasing the number of epochs. For dataset sizes of 0.05M, 0.1M, 0.25M, 0.5M, and 1M, the number of epochs is set to 90, 80, 70, 60 and 50, respectively. The warm-up period is set to 10% of the total epochs, resulting in 9, 8, 7, 6, and 5 warm-up epochs for each dataset size.

When training the ResNet50 from scratch, we use a batch size of 2048 and a learning rate of 0.2 with cosine annealing for the learning rate schedule. The weight decay is set to 1e-4. As the dataset size increases, we adjust the number of iterations to be increased while decreasing the number of epochs. For dataset sizes of 0.05M, 0.1M, 0.25M, 0.5M, and 1M, the number of epochs is set to 300, 250, 200, 150, and 100, respectively. The warm-up period is set to 10% of the total epochs, resulting in 30, 25, 20, 15, and 10 warm-up epochs for each dataset size.

B Training an image classifier from scratch

We further evaluate the dataset generation methods by training ResNet50 [68] model from scratch and evaluating it on the validation dataset of ImageNet. The results are shown in Figure C.1.

Performance improves with dataset size. Contrary to CLIP fine-tuning, training from scratch with data from the ClassPrompt method improves with increasing dataset size. Similarly, the performance of all few-shot guided generation methods also improves consistently with data scale for all k-shot settings.

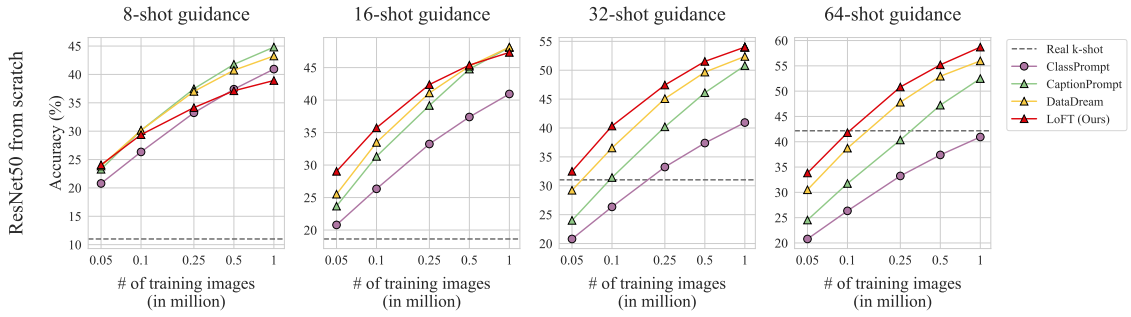


Figure C.1: Classification accuracy on ImageNet when training ResNet50 from scratch on synthetic data generated from different methods at different scales. We report few-shot guidance on 8, 16, 32, and 64 images per class and a baseline of training CLIP only on k-shot real data. LoFT benefits from a larger number of real images as guidance.

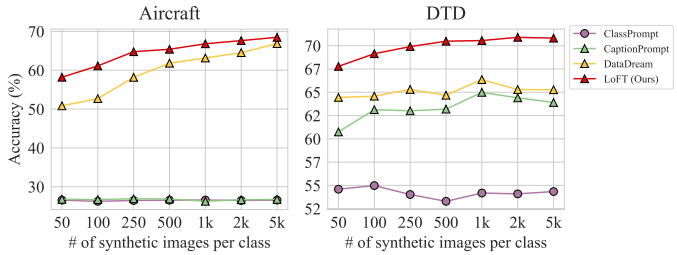


Figure C.2: Scaling the number of synthetic data on Aircraft and DTD datasets when fine-tuning CLIP.

The best method depends on the k-shot setting. The best method differs depending on the number of k-shot real images used for guidance. In the case of smaller k-shot settings (8-shot), CaptionPrompt outperforms both DataDream and LoFT. However, as the number of shots increases, DataDream and LoFT start to outperform CaptionPrompt, with LoFT consistently performing better than DataDream. For instance, LoFT achieves 58.70% at 1M scale in 64-shot while DataDream and CaptionPrompt achieve 56.00% and 52.48%, respectively.

Conclusions differ between fine-tuning and training from scratch. While the best method for training from scratch depends on the number of k-shot real images used for guidance, LoFT outperforms all baseline methods consistently across all k-shot settings when fine-tuning the CLIP model. Synthetic images by LoFT complement the prior knowledge contained in CLIP because it generates images that accurately represent the features of each class. In contrast, CaptionPrompt introduces greater diversity but since CLIP has already been pre-trained on a broad range of diverse images, it does not provide as much complementary value, limiting its effectiveness for fine-tuning.

C Scaling up to 5000 images per class on fine-grained datasets

To study the scaling ability of synthetic data size on fine-grained dataset, we conduct experiments by generating up to 5000 images per class for the Aircraft and DTD datasets. For the

few-shot synthetic data generation methods, we use 16-shot real images as guidance. The results from fine-tuning CLIP are shown in Figure C.2. ClassPrompt and CaptionPrompt reach a plateau from the beginning, indicating that increasing the number of synthetic images does not improve the classification performance. On DTD, LoFT reaches a plateau at 500 images per class, while on the Aircraft, performance continues to improve up to 5000 images per class.

D Qualitative results on fine-grained datasets

We show qualitative results of our LoFT method on Aircraft and Cars datasets. For the DHC-8-100 class on the Aircraft dataset in Figure C.10a, LoFT generate a propeller attached to the wing, which resembles real images. Moreover, for the Model B200 class in Figure C.10b, LoFT generates the shape of the class similar to real images, such as the head shape and the tail shape. Similarly in Figure C.10c and Figure C.10d, LoFT generate images of the class "Jeep Wrangler SUV 2012" and "Bugatti Veyron 16.4 Coupe 2009" that resemble the shape and fine-grained details of real images, respectively.

E Additional per-class analysis

Correlation between diversity and alignment. In addition to the recognizability and diversity metrics introduced in §4.4.3, we introduce one additional metric, **alignment**, to measure how closely the distribution of synthetic data aligns with that of real data. To quantify this, we calculate the Fréchet Inception Distance (FID) [76] score for each class, where a lower score indicates closer alignment between the synthetic and real data distributions.

As seen in Figure C.6, we observe a positive correlation between alignment and diversity for all methods. This suggests that higher diversity in the generated images come at the expense of closely mimicking the real data distribution. Moreover, the overall alignment scores are smaller for LoFT and DataDream compared to ClassPrompt and CaptionPrompt, indicating that the images generated by LoFT and DataDream align more closely with the real data distribution.

F Ratio of data points the two models disagree on the prediction

Since different methods show distinct strengths that contribute to performance gains, a natural question arises: do the classification models trained on each synthetic dataset exhibit different sets of corrected data points? To explore this, we use a ResNet50 model trained on the 0.5M-sized dataset from the ClassPrompt and 16-shot guided generation methods. We calculate the ratio of the number of data points in the validation ImageNet dataset that show inconsistent predictions relative to the total number of data points, i.e. where one model makes a correct prediction while the other model makes an incorrect one.

The results are shown in Figure C.3. Even though the three few-shot guided methods (CaptionPrompt, DataDream, and LoFT) have comparable overall accuracy (around 45% accuracy, in Figure C.1), the correction flip ratios between them are above 20%. This suggests that each synthetic dataset encourages the model to learn different features. Moreover, LoFT shows a higher flip ratio with CaptionPrompt (26.6%) compared to DataDream (20.4%). This aligns with our per-class analysis, where CaptionPrompt maintains performance by leveraging greater diversity in image distribution, while LoFT and DataDream have higher recognizability, focusing more on image fidelity.

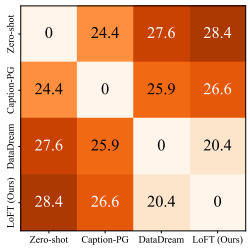


Figure C.3: Ratio of data points the two models disagree on the prediction.

G Qualitative comparison on ImageNet

We present additional qualitative results for 8 classes, i.e. Hourglass, Hard disk drive, Joystick, Weighing scale, Carved Pumpkin, Diaper, Swing, and iPod, in Figure C.7 and Figure C.8.

Taking the class Hourglass in Figure C.7a as an example, real images show hourglasses with diverse frames and varying sand colors. The images generated by ClassPrompt show less color variation. While CaptionPrompt and DataDream generate more colorful images, some of them are not easily recognizable as hourglasses. In contrast, LoFT generates images that maintain both diversity in the frame and sand color while clearly representing the hourglass.

Taking the class Swing in Figure C.8c as another example, real images show one or multiple swings, sometimes with a person riding them. Some of the generated images by ClassPrompt does not look like a swing, but rather resemble a chair. For CaptionPrompt and DataDream, some of the generated images focus more on the human subject than the swing itself, making the swing less visible. In contrast, LoFT generates clear images of swings or multiple swings, with the object clearly identifiable.

H Additional qualitative results varying λ

Figure C.4 presents examples of images generated by our LoFT method with different λ values, alongside their corresponding real images. As we adjust the weight parameter λ for the LoRA fusion, we observe distinct trends in the generated outputs. When λ is set to either 0 or 1, the generated images closely resemble the original real images. However, this approach limits the diversity of outputs across different seeds. As λ approaches 0.5, we achieve an optimal balance that enhances the diversity of the generated images while preserving their quality. Each generated image effectively integrates features from the two original real images while resembling in-distribution data. This characteristic makes the synthetic training dataset produced by LoFT beneficial for classification tasks.

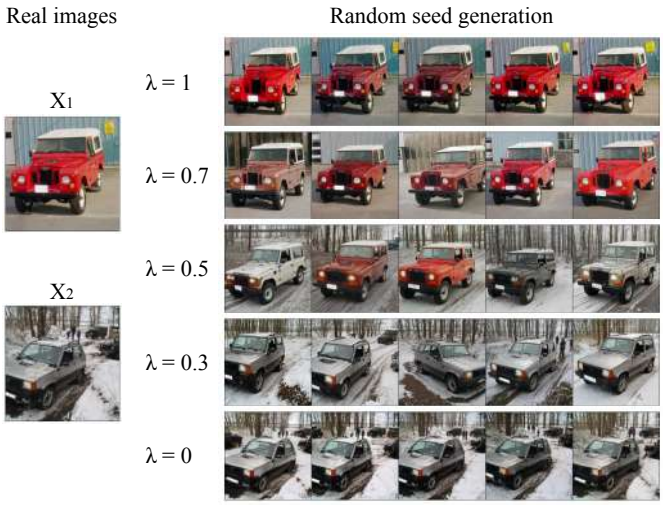


Figure C.4: Ablation study of qualitative results on λ variation when fusing LoRAs. Given two images of Jeep class, $\lambda = 0.5$ merges features from both real images while maintaining diversity with random seed image generation. As λ approaches 0 or 1, the generated images become more similar to the original image and looses diversity.

We present additional qualitative results in λ variation for 4 classes, i.e. Hourglass, Carved Pumpkin, Diaper, and Swing, in Figure C.9.

Taking the class Hourglass in Figure C.9a as an example, a real image x_1 shows a single hourglass with a wooden frame while another real image x_2 shows multiple hourglasses without a wooden frame. When $\lambda = 1$ or 0, the images generated by different random seeds closely resemble one of the real images. When $\lambda = 0.5$, the generated images show both diversity and high fidelity: some images have wooden frame while others do not, and some display multiple multiple hourglasses while others show only a single hourglass.

Taking the class Swing in Figure C.9d as another example, a real image x_1 shows a baby riding a swing colored with yellow and blow while another real image x_2 shows only a yellow swing. When $\lambda = 1$ or 0, the images generated by different random seeds closely resemble one of the real images. When $\lambda = 0.5$, the generated images show both diversity and high fidelity: the color of the swing is different, and a baby is riding a swing in some of the images.

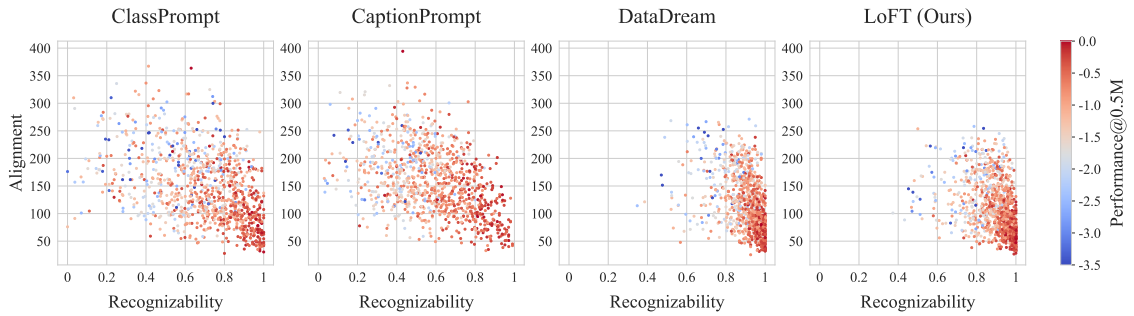


Figure C.5: Per-class analysis of recognizability and alignment in synthetic datasets generated from different methods. The color indicates a log-likelihood of the ImageNet validation dataset when CLIP is fine-tuned on the 0.5M-sized synthetic dataset in the 16-shot setting.

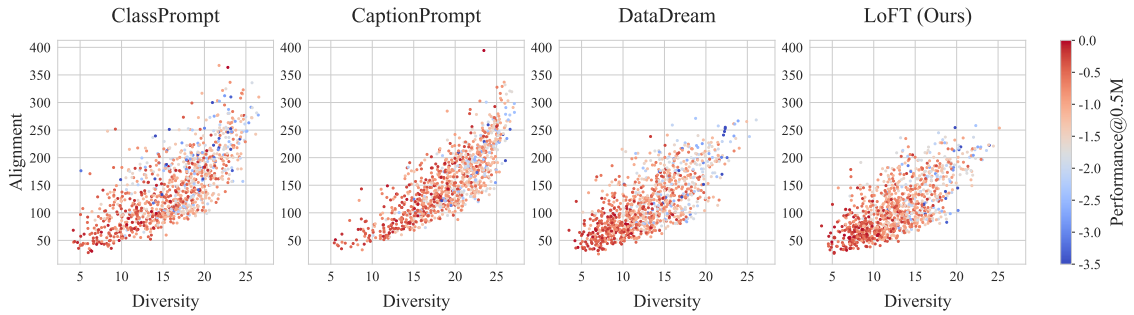
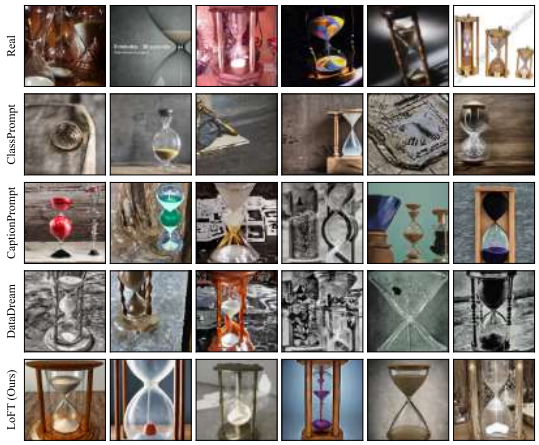


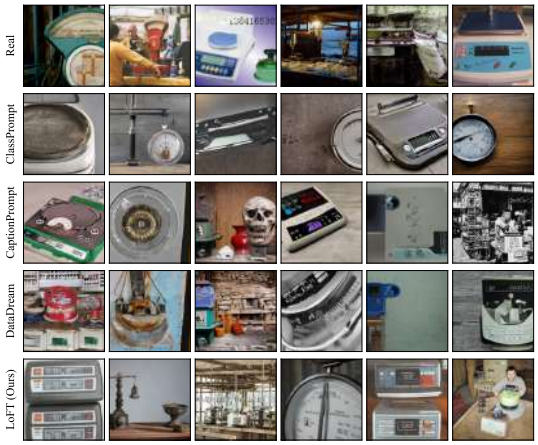
Figure C.6: Per-class analysis of diversity and alignment in synthetic datasets generated from different methods. The color indicates a log-likelihood of the ImageNet validation dataset when CLIP is fine-tuned on the 0.5M-sized synthetic dataset in the 16-shot setting.



(a) Hourglass

(b) Hard disk drive

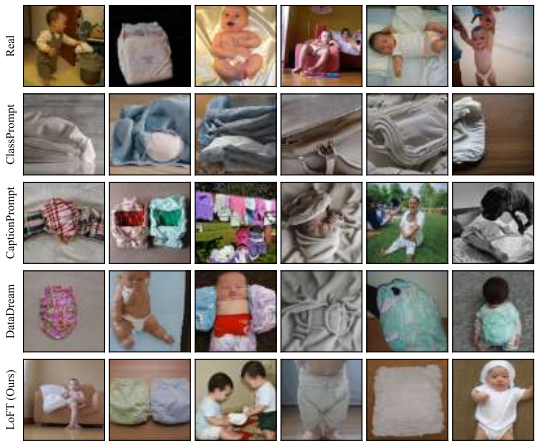
(c) Joystick



(d) Weighing scale

Figure C.7: Qualitative examples for the classes Hourglass, Hard disk drive, Joystick, and Weighing scale.

(a) Carved Pumpkin

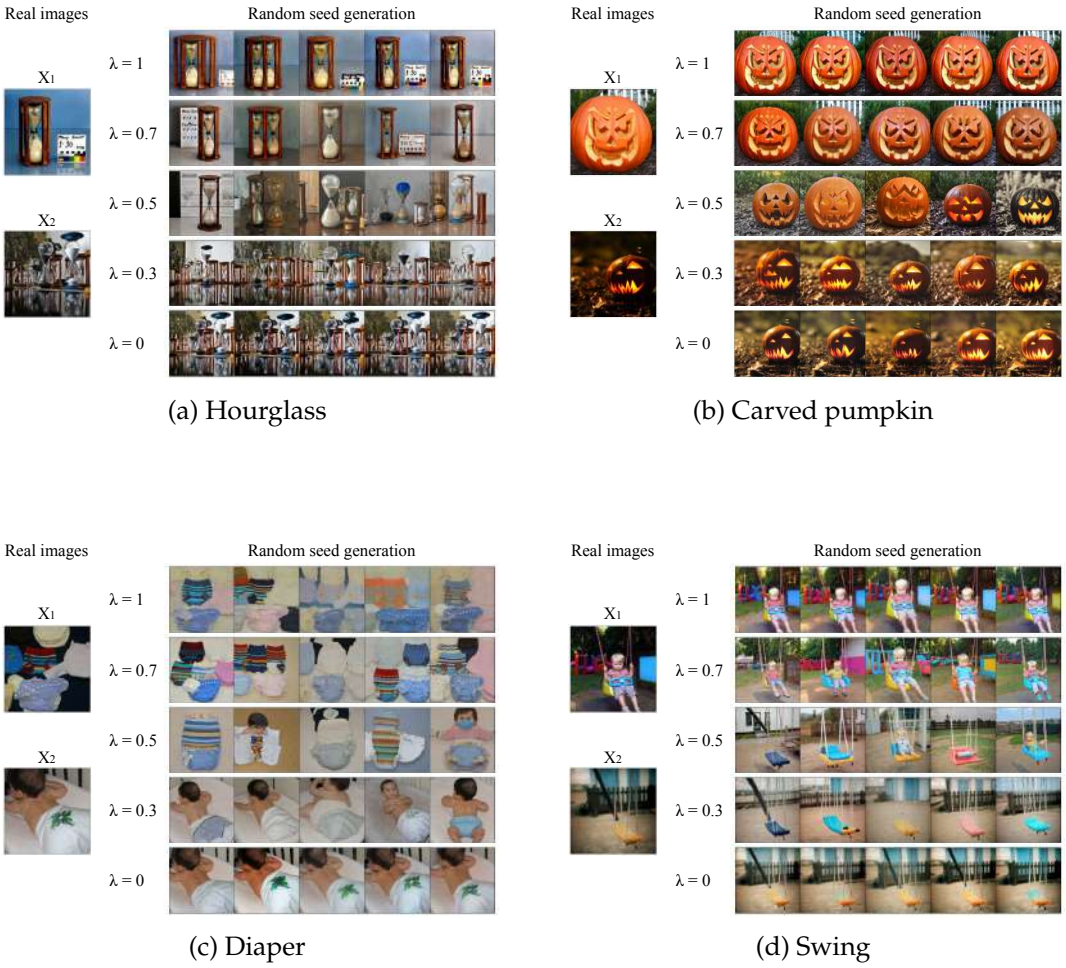


(b) Diaper

(c) Swing

(d) iPod

Figure C.8: Qualitative examples for the classes Carved Pumpkin, Diaper, Swing, and iPod.



(a) Hourglass

(b) Carved pumpkin

(c) Diaper

(d) Swing

Figure C.9: Ablation study of qualitative results on λ variation when fusing LoRAs.

(a) "DHC-8-100" class on Aircraft

(b) "Model B200" class on Aircraft

(c) "Jeep Wrangler SUV 2012" class on Cars

(d) "Bugatti Veyron 16.4 Coupe 2009" class on Cars

Figure C.10: Qualitative results of our LoFT method on Aircraft and Cars datasets.

D

Improving Intervention Efficacy via Concept Realignment in Concept Bottleneck Models

A Details of the Intervention Procedure

In Algorithm 1 we describe the standard process of performing interventions in concept-based models using an intervention policy.

Algorithm 1 Intervention Algorithm

Inputs:
    T (total number of interventions)
    π (intervention policy, which takes the concepts as input)
    $\hat{c}$ (concepts predicted by the concept encoder)
$\tilde{c} \leftarrow \hat{c}$ ▷ output of the concept encoder, g
for $t \in \{0, \ldots, T-1\}$ **do**
    $i \leftarrow \pi(\tilde{c})$ ▷ i is the concept that we want the user to intervene on
    $\tilde{c}_i \leftarrow c_i$ ▷ replace the ith concept in $\tilde{c}$ with its ground truth value c_i
end for
return $\tilde{y} = f(\tilde{c})$ ▷ updated class prediction after all interventions have been performed

In Algorithm 2 we describe the procedure used in our setup, which realigns unintervened concepts following an intervention step. We use this algorithm to compute the loss for training the realignment model.

B Comparison Between Random and UCP Policies

In this section, we compare the classification accuracies achieved by following the random and UCP intervention policies on the three datasets, respectively. In Fig. D.1 we show that the UCP policy is superior across all datasets, and is therefore our default policy across all experiments in this study.

Algorithm 2 Realignment Model Training Loss

Inputs:
    T (total number of interventions)
    π (intervention policy, which takes the concepts as input)
    $\hat{c}$ (concepts predicted by the concept encoder)
$\tilde{c} \leftarrow \hat{c}$ ▷ output of the concept encoder, g
$\kappa_{-1} \leftarrow \hat{c}$ ▷ initialize realigned concepts
$\mathcal{L} \leftarrow 0$ ▷ initialize loss
for $t \in \{0, \ldots, T-1\}$ **do**
    $i \leftarrow \pi(\kappa_{t-1})$ ▷ i is the concept that we want the user to intervene on
    $\tilde{c}_i \leftarrow c_i$ ▷ replace the ith concept in $\tilde{c}$ with its ground truth value c_i
    $\kappa_t \leftarrow u(\tilde{c})$ ▷ output of realignment model
    $\mathcal{L} \leftarrow \mathcal{L} + \text{CE}(\kappa_t, c)$ ▷ aggregate loss
end for
return $\mathcal{L}/T$ ▷ average loss across all intervention steps

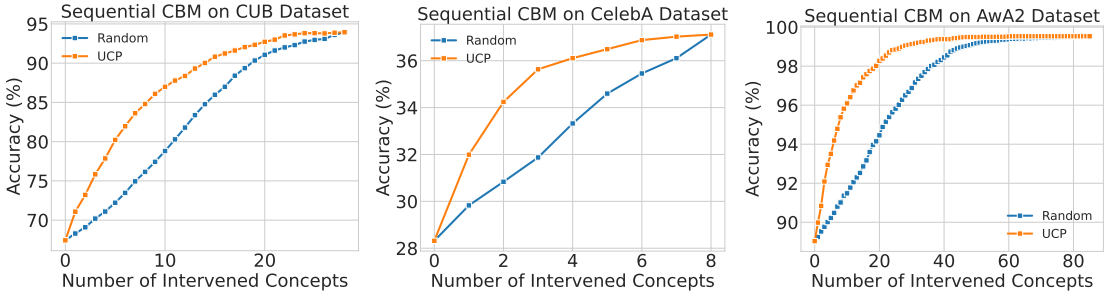


Figure D.1: Comparison between accuracy under UCP and Random intervention policies. UCP is superior in all three datasets.

C Additional Results on IntCEMs

In this section, we report the performance of posthoc concept realignment on the intervention-aware CEMs (IntCEMs) on the CelebA and AwA2 datasets to supplement the results in Section 5.4.3. In Fig. D.2 we show that concept realignment improves the performance of the SoTA approach in both datasets.

D Additional Results

Here, we report the area under the curve of concept prediction loss and classification accuracy using three different random seeds. It can be seen in Tables D.1 and D.2 that concept realignment consistently improves performance on both metrics.

Table D.1: Area Under Curve (AUC) of Concept Prediction Loss and Classification Accuracy with/without CIRM for three random seeds on the CUB dataset. We use the same backbone for sequential and independent CBMs. CIRM improves performance across all models and runs.

Base Model	**Realigned**	**Concept Loss AUC ↓**			**Accuracy AUC ↑**		
		Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
Sequential CBM	×	6.72	7.11	6.77	2460.86	2394.1	2444.08
	✓	**3.16**	**3.16**	**3.24**	**2510.48**	**2460.41**	**2501.08**
Independent CBM	×	6.72	7.11	6.77	2653.37	2652.75	2652.47
	✓	**3.16**	**3.16**	**3.24**	**2678.09**	**2675.04**	**2675.48**
Joint CBM	×	5.93	5.84	5.89	2580.28	2533.56	2591.32
	✓	**3.67**	**3.49**	**3.58**	**2608.89**	**2559.93**	**2622.53**
CEM	×	5.99	13.19	6.50	2521.31	1681.97	2579.84
	✓	**3.21**	**6.66**	**3.43**	**2558.07**	**1762.58**	**2617.25**

Table D.2: Area Under Curve (AUC) of Concept Prediction Loss and Classification Accuracy with/without CIRM for three random seeds on the CelebA dataset. We use the same backbone for sequential and independent CBMs. CIRM improves performance across all models and runs.

Base Model	**Realigned**	**Concept Loss AUC ↓**			**Accuracy AUC ↑**		
		Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
Sequential CBM	×	1.59	1.64	1.65	281.09	279.64	279.47
	✓	**1.51**	**1.53**	**1.55**	**284.76**	**284.00**	**284.21**
Independent CBM	×	1.59	1.64	1.65	280.86	308.38	310.57
	✓	**1.51**	**1.53**	**1.55**	**282.48**	**312.72**	**316.45**
Joint CBM	×	2.88	3.23	3.10	273.06	236.22	296.80
	✓	**1.75**	**1.77**	**1.74**	**273.76**	**246.09**	**303.76**
CEM	×	1.65	1.90	1.83	396.70	366.87	361.60
	✓	**1.49**	**1.66**	**1.58**	**401.84**	**370.88**	**363.57**

Table D.3: Area Under Curve (AUC) of Concept Prediction Loss and Classification Accuracy with/without CIRM for three random seeds on the AwA2 dataset. We use the same backbone for sequential and independent CBMs. CIRM improves performance across all models and runs.

Base Model	**Realigned**	**Concept Loss AUC ↓**			**Accuracy AUC ↑**		
		Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
Sequential CBM	×	4.26	3.76	3.92	8363.9	8411.17	8373.00
	✓	**1.13**	**1.2**	**1.15**	**8397.79**	**8437.69**	**8400.66**
Independent CBM	×	4.26	3.76	3.92	8403.45	8410.39	8407.77
	✓	**1.13**	**1.2**	**1.15**	**8437.31**	**8437.59**	**8438.71**
Joint CBM	×	4.77	4.34	4.47	8276.37	8350.96	8346.31
	✓	**1.5**	**1.54**	**1.51**	**8326.95**	**8391.89**	**8389.76**
CEM	×	4.9	4.04	3.92	8429.35	8438.99	8439.87
	✓	**1.69**	**1.45**	**1.46**	**8433.38**	**8439.9**	**8439.52**

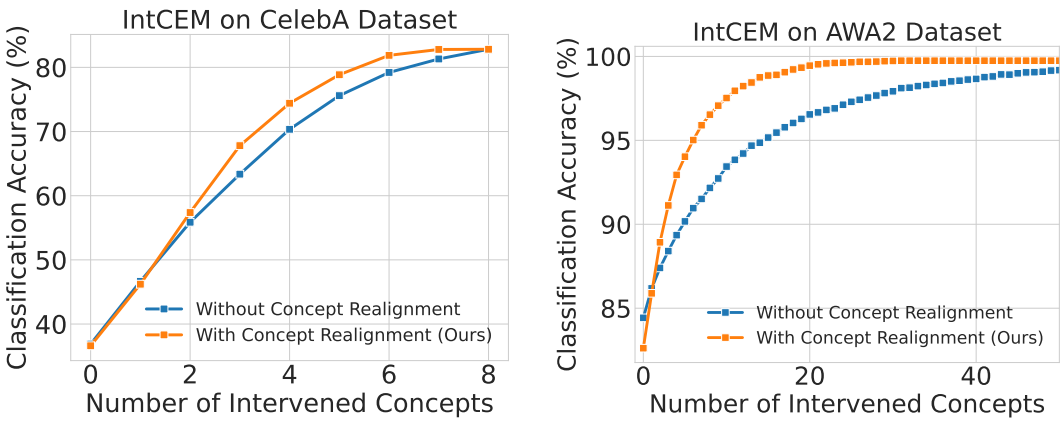


Figure D.2: Classification accuracy with and without posthoc concept realignment in intervention-aware CEMs. In both cases, concept realignment improves performance of the base IntCEM model.

E

Publications and contributions

A Publications

This thesis is based on the following publications. An overview of the contributions can be found in Sec. 1.3. Asterisks (*) indicate shared first author publications. Bold names correspond to the name of the author of this thesis.

1. [162] K. Roth *, **J. Kim** *, A. Kopeke, O. Vinyals, C. Schmid, Z. Akata, "Waffling Around for Performance: Visual Classification with Random Words and Board Concepts". In: *IEEE / CVF International Conference on Computer Vision (ICCV)*. 2023.

2. [88] **J. Kim**, S. Alaniz, C. Schmid, Z. Akata, "Feasibility with Language Models for Open-World Compositional Zero-Shot Learning". In: *Workshops of the European Conference on Computer Vision (ECCV-W)*. 2024.

3. [189] N. Singhi, **J. Kim**, K. Roth, Z. Akata, "Improving Intervention Efficacy via Concept Realignment in Concept Bottleneck Models". In: *the European Conference on Computer Vision (ECCV)*. 2024.

4. [90] **J. Kim** *, J. Bader, S. Alaniz, C. Schmid, Z. Akata, "DataDream: Few-shot Guided Dataset Generation". In: *the European Conference on Computer Vision (ECCV)*. 2024.

5. [89] **J. Kim**, S. Alaniz, C. Schmid, Z. Akata, "LoFT: LoRA-Fused Training Dataset Generation with Few-shot Guidance". In: *Under review*. 2025.

The following publications were done during the course of the PhD, and are also closely related to the key objectives, but they are not part of this thesis to have a more coherent flow.

1. [92] Y. Kim *, **J. Kim** *, Z. Akata, J. Lee, "Large Loss Matters in Weakly Supervised Multi-Label Classification". In: *IEEE / CVF on Computer Vision and Pattern Recognition Conference (CVPR)*. 2022.

2. [93] Y. Kim, **J. Kim**, J. Jeong, C. Schmid, Z. Akata, J. Lee, "Bridging the Gap between Model Explanations in Partially Annotated Multi-label Classification". In: *IEEE / CVF on Computer Vision and Pattern Recognition Conference (CVPR)*. 2023.

3. [91] **J. Kim**, A. Kopeke, C. Schmid, Z. Akata, "Exposing and Mitigating Spurious Correlations for Cross-Modal Retrieval". In: *IEEE / CVF Workshops on Computer Vision and Pattern Recognition Conference (CVPR)*. 2023.

4. [202] U. Upadhyay, **J. Kim**, C. Schmid, B. Schölkopf, Z. Akata, "Likelihood annealing: Fast calibrated uncertainty for regression'. In: *arXiv preprint*. 2023.

B Contributions

This section presents the contributions of the authors for the publications included in this thesis.

Waffling around for Performance: Visual Classification with Random Words and Broad Concepts. This work was done in collaboration with Karsten Roth, Sophia Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. Karsten Roth and Jae Myung Kim were co-first authors of the paper. Both contributed extensively to the development of ideas, implementation of the codebase, and writing a paper, with Karsten Roth contributing slightly more overall. Sophia Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata had a supervisory role throughout the project to discuss ideas and write a paper.

Feasibility with Language Models for Open-World Compositional Zero-Shot Learning. This work was done in collaboration with Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. Jae Myung Kim was the first author and contributed to developing the idea, running the experiments, and writing a paper. Stephan Alaniz, Cordelia Schmid, and Zeynep Akata had a supervisory role throughout the project to discuss ideas and write a paper.

DataDream: Few-shot Guided Dataset Generation. This work was done in collaboration with Jessica Bader, Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. Jessica Bader and Jae Myung Kim were co-first authors of the paper. Both contributed extensively to the development of ideas, implementation of the codebase, and writing a paper, with Jae Myung Kim contributing slightly more to the first two, and Jessica Bader contributing slightly more to the writing. Stephan Alaniz, Cordelia Schmid, and Zeynep Akata had a supervisory role throughout the project to discuss ideas and write a paper.

LoFT: LoRA-fused Dataset Generation with Few-shot Guidance. This work was done in collaboration with Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. Jae Myung Kim was the first author and contributed to developing the idea, running the experiments, and writing a paper. Stephan Alaniz, Cordelia Schmid, and Zeynep Akata had a supervisory role throughout the project to discuss ideas and write a paper.

Improving Intervention Efficacy via Concept Realignment in Concept Bottleneck Models. This work was done in collaboration with Nishad Singhi, Karsten Roth, and

Zeynep Akata. Nishad Singhi was the first author and contributed to running the experiments and writing a paper. Nishad Singhi, Karsten Roth, and Jae Myung Kim equally contributed to developing ideas. Zeynep Akata had a supervisory role throughout the project to discuss ideas and write a paper.