UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Enhancing Few-Shot Learning with Large Language Models

Author:
Joel DIÉGUEZ VILÀ

Supervisors:
Dra. Petia RADEVA
Javier RÓDENAS CUMPLIDO

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

June 30, 2025

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Enhancing Few-Shot Learning with Large Language Models

by Joel DIÉGUEZ VILÀ

Recently, Few-Shot Learning has gained significant momentum in the machine learning community. This field focuses on enabling models to learn from extremely limited data, often just a handful of examples per class. Unlike traditional deep learning, which relies on large-scale datasets, few-shot learning requires novel, efficient strategies that challenge conventional assumptions and fundamentally shift the paradigm toward "learning to learn", for faster, more adaptable models.

In this work, we explore the most common approaches to few-shot learning and introduce our own method. Building upon the SemFew framework, we propose a metric-based meta-learning approach using Prototypical Networks, enhanced with a semantic support module. This module uses class descriptions from WordNet, refined through a Large Language Model, to provide high-quality semantic embeddings that guide the model in understanding novel classes. Our proposed model is remarkably simple yet highly effective, achieving competitive performance with state-of-the-art methods, specially in 1-shot scenarios (only one example per class).

We validate our method across three widely used few-shot classification benchmarks: CIFAR-FS, FC100, and MiniImageNet. The results consistently demonstrate the effectiveness of incorporating semantic guidance to face unseen classes. Furthermore, we present an in-depth study of modern LLMs, evaluating their performance across different prompting strategies, and investigating multiple sources of data for generating the best semantic representations. This analysis offers valuable insights into how semantic guidance can be optimized for few-shot learning.

Overall, this work demonstrates the power of combining simple metric-based learning with rich semantic embeddings, offering a practical and competitive alternative to more complex architectures while encouraging new directions for future research in few-shot learning.

The source code is available at: https://github.com/jdieguvi15/TFM-SemFew.

Acknowledgements

During the course of this work, I have received invaluable support from many people, both academically and personally, and I would like to take this opportunity to express my gratitude to them here.

To Petia Radeva, my main supervisor, for proposing this project and making her outstanding team available to me, whose support has been invaluable. I would also like to thank her for organizing the exchange at NPU University in Xi'an, where I had the chance to deepen my knowledge of LLMs.

To Javier Rodenas, for being there every week, always offering support, sharing insightful ideas, and challenging my own with his expertise in the field.

To Eduardo Aguilar, for his contributions and guidance as a remarkable researcher in this area.

To my family and friends, who have been by my side throughout the master's and have given me so many wonderful moments. Especially to Clàudia, Alba, Alejandro and Júlia, for their constant encouragement and for always making me smile.

Thank you all so much.

Contents

Chapter 1

Introduction

1.1 Motivation

In recent years, the exponential growth of digital life and online activity has generated a massive amount of publicly available data from different sources like social networks, multimedia platforms, mobile devices, and distributed sensors. This phenomenon has enabled the creation of large annotated datasets such as ImageNet, which contains over 14 million images classified into more than 20,000 categories, and has served as a turning point in the evolution of deep learning. Thanks to the availability of this large-scale data, deep neural network models have achieved unprecedented advances. Models like the Transformers, which are highly dependent on huge volumes of data to function properly, have revolutionized many fields such as natural language processing and computer vision. This phenomenon has consolidated an approach to machine learning based on big data and intensive computation.

However, this trend contrasts with a fundamental limitation in many real-world scenarios: the scarcity of annotated data. Although the total volume of data generated is very high, the data available in each specific domain or per individual device is often very limited, with only a few examples per class or scene. This phenomenon is known as the "long tail" effect, where most classes or environmental conditions are poorly represented in the dataset. For example:

- In medicine, rare disease diagnosis have an extremely low number of samples per category.
- In language processing, many dialects, indigenous languages or specialized contexts have little data available.
- In computer vision, the detection of rare animals or unusual scenes.

Even if we have lots of data, it is often not applicable to specific domains where collecting labeled data is difficult, costly, or simply not feasible. Therefore, the need arises to develop artificial intelligence systems that are able to learn from limited data and successfully identify it.

Few-Shot Learning (FSL) is the branch of machine learning that seeks to tackle this problem. Its concept is inspired by the robust reasoning and analytical capabilities of humans: we are capable of recognizing a new class or object with only a few examples, thanks to our ability to abstract relevant features, compare patterns and apply previous knowledge in order to learn from them.

Moreover, today's most powerful AI models, Large Language Models, can effortlessly generate detailed class descriptions. We could explore the idea of using these information to significantly simplifying the process of learning to recognize new concepts.

1.2 Context of the Project

In early work on few-shot learning for image classification, researchers focused primarily on visual-based approaches [Dong et al., 2022b, Hao et al., 2023], extracting features directly from image data. Despite many successes in this direction, visual-only models struggle when the only data support for a class are periphery samples, specially if there is only one, i.e. in one-shot scenarios. Hence, the idea of applying auxiliary information to help the model better understand the data.

From this perspective, the field began to incorporate semantic information as an auxiliary source of knowledge. Instead of relying only on what the image looks like, we add some textual information to tell the model something about what the class actually means. In the beginning, some works used just the class names [Chen et al., 2023b, Yan et al., 2021, Wang et al., 2019], however, just a name is not enough to understand a new class. Even for humans, if a person never saw a zebra, just the name would not be enough to be able to identify them, they would need a definition like "a horse with black and white stripes". Moreover, some class names like "organ" or "ear" are polysemous, which can further confuse the model. Therefore, it is very important to collect high-quality semantics.

That is where Large Language Models (LLMs) come into play. It would be very consuming to manually write high-quality definitions to all of the different classes. Furthermore, few-shot learning requires the fast adaptation to new unknown classes, thus, we need a way to automate the process. LLMs have shown strong capabilities in understanding and generating natural language: generate comprehensible text, analogies, paraphrase, and they are easily adaptable to new domains, making them a perfect tool for this task.

This approach has already been attempted in the paper *Semantic-Aided Few-Shot Learning (SemFew)* [Zhang et al., 2024], which demonstrated state-of-the-art performance in few-shot classification. Building on this foundation, the goal of this thesis is to enhance the effectiveness of semantic guidance by carefully designing and optimizing the prompts used to generate class descriptions, as well as introducing some modifications to their integration into the model.

1.3 Objectives

The main objective of this thesis is to explore how Large Language Models can be used to generate high-quality semantic representations of visual classes, with the aim of improving few-shot learning performance. Specifically, our objectives are:

- Study the state-of-the-art in few-shot learning, establishing a foundation for analysis and meaningful comparison of results.
- Analyze the approach proposed by SemFew, understand its strengths and limitations, and propose new paths for improvement.
- Design and evaluate different prompt formulations for LLMs to produce the most informative and precise class descriptions.
- Validate the results across different classification tasks and datasets.

1.4 Contributions

This thesis makes several key contributions to the field of few-shot learning:

- Extensive analysis on the different possible prompt formulations and LLMs, and their resulting definitions, quantitatively and qualitatively.
- A new proposal for prompting to improve the class definitions used in previous works.
- An accuracy improvement of SoA of over 1% in few-shot testing over prior approaches using similar architectures.
- Validation of results in multiple datasets widely used for few-shot learning: miniImageNet, CIFAR-FS, and FC100, demonstrating the robustness and generalizability of our approach.

1.5 Organization of the Report

This report is structured into five main chapters:

1. **Introduction** introduces the motivation, context and objectives behind this work, and highlights its main contributions.
2. **Scientific Background** presents the knowledge required to understand the thesis. It begins with the most focal research topic of the thesis: Few-Shot Learning (FSL), its differences from traditional machine learning and the principal approaches used in the field. Then, we cover other topics relevant to the thesis such as Large Language Models (LLMs) and machine learning ensembles.
3. **Methodology** describes the proposed model, how is its architecture composed, how we get the semantics for the classes and how to integrate them with the visual features.
4. **Validation** presents the validation process and experimental results. It explains the experimental setup: datasets used, evaluation metrics, ... and reports the main findings, along with an ablation study to study the impact of individual components.
5. **Discussion** reflects on the results, analyzes the advantages and limitations of the approach. It also presents possible future lines of development and reflects on the future for few-shot learning.

Finally, in the annexes, we include additional figures and results that further illustrate and complement the main findings of our work.

Chapter 2

Scientific Background

The Scientific Background chapter provides an overview of the basic concepts, methodologies and challenges required to understand the project. It is intended for readers with basic notions of machine learning, including the basics of supervised learning, neural networks and training paradigms such as transfer learning. If the reader needs a more detailed explanation of these topics, they can refer to my previous work [Diéguez, 2023].

In this section, we begin by formally introducing the main topic of this thesis: the few-shot learning problem, including its task formulations, motivations, and different approaches to tackle it. Following this, we explore other necessary topics of machine learning that will be used for the development of the proposed model, such as Large Language Models, Prompt Engineering, and Ensembles.

2.1 Few-Shot Learning

Few-Shot Learning (FSL) is an advanced technique within the field of Machine Learning that allows an artificial intelligence model to learn to make accurate predictions from a very limited number of labeled examples. Unlike traditional approaches, which require large volumes of data to achieve good results, FSL tries to optimize the generalization capacity of the model from only a few examples per class.

This paradigm is based on learning general representations (i.e. patterns, abstract features and latent structures), which can be transferred to new tasks with little or no additional retraining. This means that the model not only learns to classify certain classes, but also to extract the features that allow it to identify new classes and differentiate them. In other words, it develops a deeper understanding of the structure of the problem, which allows it to adapt quickly to unfamiliar contexts [Tyukin et al., 2021].

2.1.1 Differences with traditional Machine Learning

The main differences between traditional Machine Learning and Few-Shot Learning are:

- **Amount of data**: traditional learning requires a large amount of data (thousands or millions of examples per class), while few-shot works with very small datasets (as few as 1–10 examples per class).

- **Inspiration**: traditional learning is based on statistical modeling and curve fitting, while few-shot is inspired by human cognition and fast pattern recognition from limited examples.

- **Model architecture**: traditional training typically uses simpler, task-specific architectures, while few-shot learning relies on more sophisticated architectures such as meta-learning or siamese networks, spcialized for fewer data.
- **Task Adaptability**: traditional learning often needs significant retraining or fine-tuning for a specific task, while few-shot can quickly adapt to new tasks without retraining the entire model. For example, adding a new class of rare animal to a typical image classification model would require lots of retraining but few-shot would be prepared for it.

2.1.2 Problem formulation

Based on the commonly used formulation proposed in [Wang et al., 2020].

Let C_{base} be a set of base classes with a large number of labeled examples, and C_{novel} a disjoint set of novel classes, where only K labeled examples per class are available. The objective is to generalize from C_{base} to C_{novel}.

The structure of FSL training and evaluation generally follows the N-way K-shot framework, where:

- N is the number of classes in a task.
- K is the number of labeled examples (or "shots") per class.

For example, in 3-way-2-shot we classify images among 3 classes based on 2 examples given from each class, like in Figure 2.1. The most common cases are 5-way-5-shot, 5-way-1-shot (typically called *one-shot*), and 5-way-0-shot (called *zero-shot*). But this last one is quite different, as it does not use supporting images, thus it requires different strategies.

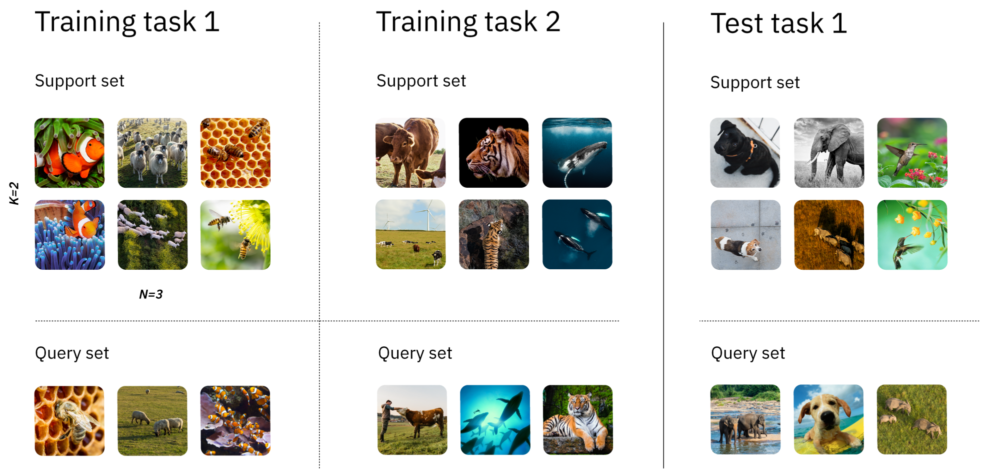


FIGURE 2.1: The few-shot task example for 3-way 2-shot.

In N-way-K-shot classification, the model undergoes multiple episodes of training, each consisting of one or more training tasks. Likewise, the model is evaluated via test tasks, following the same structure. Each training and test episode comprises disjoint subsets:

- The **support set** $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N \cdot K}$, which contains K labeled training samples for each of the N classes. The model uses this data to learn representations or prototypes for each class.
- The **query set** $\mathcal{Q} = \{(x_j, y_j)\}_{j=1}^{M}$, which contains unseen examples from the same N classes. The model is evaluated using a loss function $\mathcal{L}$ based on its ability to correctly classify these examples using only the information learned from $\mathcal{S}$.

The model is trained over multiple episodes, each consisting of different tasks sampled from C_{base}.

As out objective it to generalize well to unseen data, rather than focusing on recognizing any specific classes, each training task includes different classes.

During each episode, the model minimizes a loss function over the query set:

$$\mathcal{L}_{\mathcal{T}} = \frac{1}{|\mathcal{Q}|} \sum_{(x,y) \in \mathcal{Q}} \ell(f_\theta(x; \mathcal{S}), y)$$

where f_θ is the few-shot model parameterized by θ, which uses information from $\mathcal{S}$ to make predictions on $\mathcal{Q}$, and ℓ is the loss function (normally cross-entropy). With the goal to optimize the parameters θ that minimizes the expected loss over query sets from new tasks $\mathcal{T}$:

$$\min_{\theta} \mathbb{E}_{\mathcal{T}_i} \left[\mathcal{L}_{\mathcal{T}_i}(f_\theta, \mathcal{S}_i, \mathcal{Q}_i) \right]$$

An important aspect of FSL is that the model is evaluated on test tasks built from entirely unseen classes, i.e. not used during training. This ensures the model's ability to generalize, rather than memorize specific classes.

2.1.3 Approaches to FSL

There are a few different approaches to tackle few-shot learning [Song et al., 2022, Tsoumplekas et al., 2025]. The main ones are Data Augmentation, Transfer Learning, Meta-Learning, Multimodal Learning, and Continual Learning. Each category offers different assumptions and strategies to overcome data scarcity. Let us see them in detail in the following.

1. Data Augmentation

Data augmentation refers to techniques that synthetically increase the number of training examples by generating new data points from existing ones. Consequently, we "solve" the small dataset by adding artificially generated new data.

It normally consists in applying various transformations (e.g. rotations, cropping, color scale changes, ...) to the pre-existing data, as shown in Figure 2.2. Therefore, increasing the size and diversity of the training dataset, which also helps to avoid overfitting.

However, it also has some disadvantages. It is very limited by the diversity of the original dataset. Thus, with very few data, adding more variations of it may cause some risk of overfitting. Moreover, there are some domains where transformations are hard (e.g. in medical imaging).

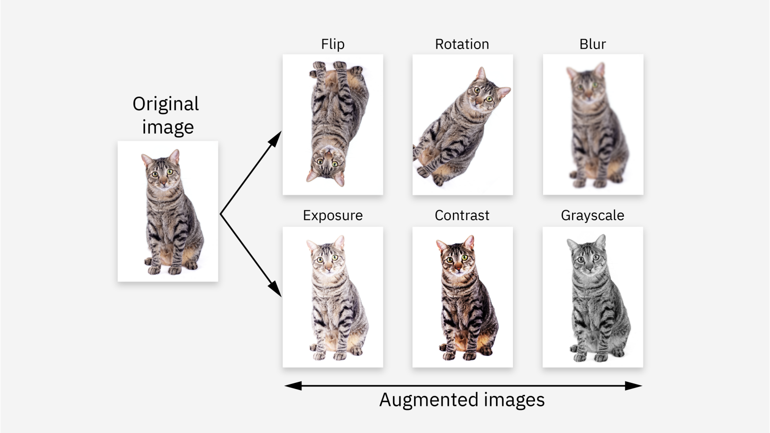


FIGURE 2.2: Data Augmentation applied to the image of a cat.

2. Transfer Learning

Transfer Learning is a property of ML models that consists on using knowledge learned from a task for a different related task in order to boost its performance. For example, the knowledge gained from training a model to recognize cats on images can greatly help to learn to recognize dogs in images.

It is based on the principle that the base of the model, i.e. extracting the main features of the data, is quite similar among different related tasks. Thus, it us useful for all of them. The later layers specified on the particular task.

Transfer learning plays a crucial role in FSL. A common and effective approach is to first train a model on a large-scale dataset (such as ImageNet for image classification), to learn how to extract features. This results in a strong backbone that captures general useful representations to various tasks and domains. Once the backbone is trained, the model can be adapted to a new task involving a small number of examples by training only the final layers on top of the frozen features.

Moreover, the pretraining can be effectively carried out using Self-supervised learning (SSL), which enables the model to learn general-purpose representations without the need for manual labeling (e.g. contrastive learning methods like SimCLR [Chen et al., 2020], MoCo [Majumder et al., 2021], DINO [Zhang et al., 2022]).

3. Meta-Learning

Meta-learning, also known as “learning to learn”, is a learning paradigm that consists on training on different tasks in an episodic training mechanism. Each episode simulates a different unseen task, so the model has to learn a large number of more general auxiliary tasks, building high-level strategies or initialization states to adapt to all tasks. During this process, the model samples both the data and task space, developing an abstract learning ability. Therefore, it seeks to train a model not just to perform a specific task, but to rapidly adapt to a distribution of new tasks, which makes it more flexible and efficient in low-resource scenarios. Figure 2.3 shows an example of meta-learning training.

Meta-learning is not the same as FSL. Few-shot learning is the goal it is aming at, and meta-learning is a strategy to get there. Although, it is currently considered the best approach to address the FSL problem.

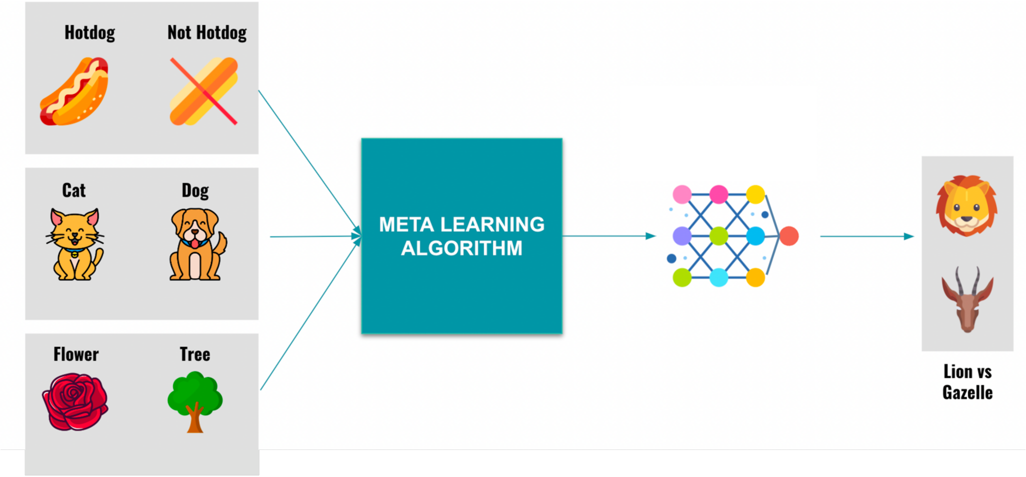


FIGURE 2.3: Graphical representation of meta-learning. The model is trained on different tasks per episode and then evaluated on differentiating lions vs. gazelles.

One could say that meta-learning is more oriented to future tasks instead of optimizing the current one. However, it is only effective when the testing and training tasks are relatively similar and it is very dependent on the network structure. There will always be a domain gap between training and testing. So it is important to properly define the model structure and the task selection per episode.

There are three main approaches to meta learning: metric-based learning, model-based learning and optimization-based learning.

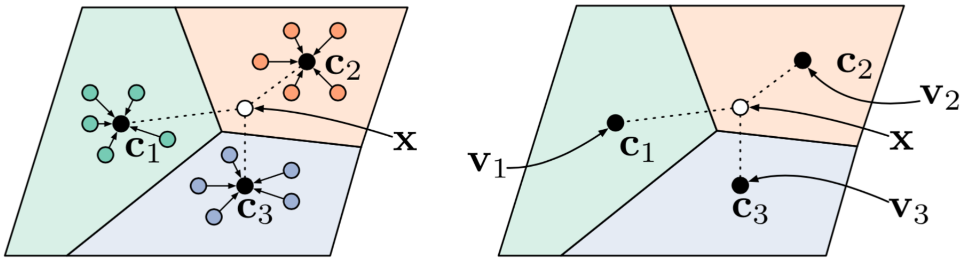
Metric-based

This approach focuses on learning an embedding space that captures the main important parts of the data and use a distance metric, predefined or learned, to measure the similarity between pairs of data points. The feature representation extractor can be the same for all tasks: Task-Invariant Representations or tailored for each: Task-Specific Representations.

Some examples of **Task-Invariant Representations** are:

- **Siamese Networks** [Koch et al., 2015], where both the support and the query are encoded using the same exact network and the matching is done between each query sample and each support sample.
- **Matching Networks** [Vinyals et al., 2016], which employ different embeddings for support and query. Then, the prediction is calculated using an attention mechanism using cosine similarity between query and support embeddings.
- **Prototypical Networks** [Snell et al., 2017], based on learning feature embeddings that can be clustered to form a single class prototype representation (like calculating the mean). Then, the query samples are classified based on proximity to the class prototypes. Figure 2.4 illustrates the process.
- **Relation Networks** [Sung et al., 2018] are similar to Prototype networks but instead of using a nearest class prototype algorithm, they use an MLP.

On the other hand, some examples of **Task-Specific Representations**:



(a) Few-shot

(b) Zero-shot

FIGURE 2.4: Graphical representation of Prototypical Networks. In the few-shot scenario, the prototype is generated with the mean of the support samples. In the 0-shot, it is extracted from external data not coming from the support

- **TADAM** [Oreshkin et al., 2018] introduced modulation on the feature extractor based on the task using learnable shift and scale vectors per convolutional layer. Moreover, the loss function incorporates learnable metric scaling to handle different distance metrics.

- **FEAT** [Ye et al., 2020] uses self-attention to contextualize each support sample relative to others in the set.

- **SSFormers** [Chen et al., 2023a] is transformer-based method that contextualizes local image features both within an image and across other support set images. It detects the most relevant patches based on the task and uses sparse attention over them to improve the matching.

- Other works like [Hu et al., 2022a] and [Hu et al., 2022b] propose to finetune the backbone for each task individually using different methods like cross-sample attention, background patch fusion or masked segmentation training, with the objective of having more separable and compact feature representations by enhancing intra-class coherence.

Optimization-based

Optimization is a crucial part of any branch of deep learning. In this case, as we are facing different tasks with very limited data, there is no guarantee that we will reach an optimal solution. Optimization-based meta-learning aims to learn optimization procedures to improve the optimization process across different tasks without multiple iterations or large volumes of data. It consists on "learning how to optimize". It can be categorized into three types: learning the update rule, learning the initial parameters, and learning to control the update rule.

- **Learning the update rule**. Inspired by the principle that learned features are better than hand-crafted ones, they replace hand-designed optimizers, like Stochastic Gradient Descent (SGD) [Ruder, 2017], with learned update rules. This is achieved by casting the design of the rule as a learning problem.

  [Andrychowicz et al., 2016] proposes to use an LSTM (Long Short-Term Memory RRN) to control the update steps. LSTMs are suitable for this since they maintain past information in their internal state, allowing for previous updates to inform next ones. Meta-Learner LSTM from [Ravi et al., 2016] extends this by using LSTMs for both initialization and update strategy. However, these methods do not scale well and have trouble capturing any dependencies between different model parameters. Instead, [Li et al., 2017a] formulates the optimization as a Reinforcement Learning (RL) problem to learn update policies via guided policy search.

- **Learning the initial parameters**. They focus on learning initial parameters that converge to optimal parameters with only a few iterations.

  The famous Model-Agnostic MetaLearning (MAML) [Finn et al., 2017] was the first work that aimed to tackle this. It proposes to cast meta-learning as bi-level optimization, in two different levels. The inner loop adapts the initial parameters to the task at hand and the outer loop optimizes the initial parameters using the errors across training tasks. Therefore, it captures the shared information across tasks in the initial state, simplifying as much as possible the optimization within each task.

  After this, many models tried to improve it. MAML++ [Antoniou et al., 2019] added an outer-loop loss dependent on the inner-loop loss, training stabilization and other improvements. [Behl et al., 2019] introduced hypergradient descent. REPTILE [Nichol et al., 2018] used first-order approximation of MAML that simplified the calculation by avoiding second-order derivatives. [Rusu et al., 2019] proposed different initialization for different tasks.

- **Learning to control the update rule**. Just learning to update the rule can sometimes lead to non-convergence and they lack the main inductive bias of gradient descent. Therefore, this methods focus on learning how to control the update of a known optimizer. For example, [Li et al., 2017b] proposed to adapt MAML by also learning the learning rates and learning directions for each parameter. However, this also doubles the number of parameters, increasing the computational complexity. [Lee et al., 2018] proposed to learn which parameters are shared across tasks and use task-specific masks to select them.

Model-based

In the model-based approach, the meta-learner learns to develop the model's architecture in such a way that it can quickly adapt it to new tasks. There are two main categories: Architecture-based and Memory-based.

- The Architecture-based approach uses an "internal memory". In the beginning, RRNs were used [Duan et al., 2016], using the internal state to generate models and adapting the problem as reinforcement learning to quickly adapt to new tasks while remembering previous general data. Or Gated Recurrent Units (GRU), with a fast learner that learned to adapt to each task while the slower optimizes the policy across tasks.

  Later, more complex and general architectures were proposed. SNAIL [Mishra et al., 2018] combines temporal convolutions with self-attention to learn from sequences of experience. This allows us to attend to relevant patterns over longer time scales.

- Memory-based methods use external memory module to retain relevant information related to rare classes. For example, Meta Networks [Munkhdalai et al., 2017] uses an external memory to store "fast weights", which are temporary, task-specific parameters for rapid adaptation across tasks. It consists on a meta-learner that generates these weights, which allow the base learner to quickly adapt to each new task. Other approaches rely on external memories to store refined sample representations. Memory-Augmented Neural Networks (MANNs) [Santoro et al., 2016] store sample representations in memory and retrieve them using soft attention.

4. Continual Learning

Continual learning, also known as "lifelong learning", addresses the challenge of learning from a stream of new tasks or data distributions over time while retaining previously acquired knowledge. It has to maintain a balance between stability (preserving past knowledge) and plasticity (learning new information), a core dilemma known in continual learning as the **stability-plasticity dilemma**. Unlike traditional machine learning, where all data is accessible at once, continual learning is based on incremental updates without catastrophic forgetting, i.e. the model does not worsen its performance on earlier tasks when learning new ones. This is particularly useful in real-world scenarios where data often arrives in small batches and models must learn from only a few examples per task. Continual learning allows systems to adapt continuously and efficiently, without retraining from scratch whenever new information becomes available, as shown in Figure 2.5.

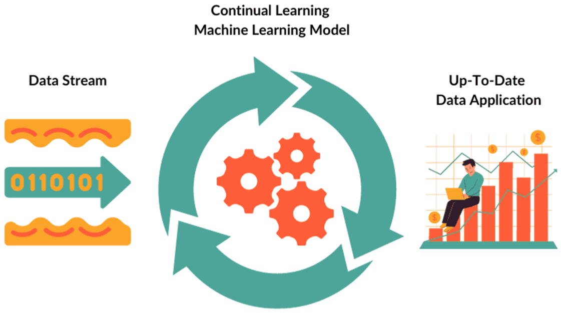


FIGURE 2.5: Continual Learning diagram

Several techniques have been developed for continual learning. **Regularization-based** methods, such as Elastic Weight Consolidation (EWC) [Kirkpatrick et al., 2017], penalize changes to model parameters that are important for previous tasks, thus preserving past knowledge. **Replay-based** approaches use a memory buffer to store and replay a subset of previous data during training on new tasks, ensuring that past experiences are revisited and not forgotten.

In contrast, **dynamic-architecture-based** methods expand or adapt the model's structure to include new tasks without interfering with previously learned ones. Progressive Neural Networks [Rusu et al., 2016], allocate a new neural network column for each incoming task. Instead of modifying the parameters of previous tasks, the new column connects laterally to earlier columns through frozen weights. Therefore, the model is able to transfer useful knowledge while preventing interference. This design ensures that previously learned tasks remain intact while still enabling the system to adapt to new information.

While meta-learning and continual learning have a very similar objective: learning on many different tasks, they differ fundamentally in their assumptions. Meta-learning focuses on learning to quickly adapt to new tasks by learning shared patterns across tasks during training and continual learning deals with a dynamic and ongoing stream of tasks, where the same model accumulates knowledge over time and retains it across tasks.

The main advantages of continual learning are real-time adaptability and no need for retraining from scratch. But they also present some disadvantages: sensitivity to task order, limited memory capacity for replay methods, and the inherent difficulty of simultaneously learning new tasks and preserving performance on old ones.

5. Multimodal Learning

Multimodal learning enhances few-shot learning by integrating information from multiple data modalities, such as text, images, and audio, to compensate for the lack of samples. This approach is inspired by human learning, where knowledge is rarely acquired through a single sense. Instead of relying only on limited visual examples, multimodal FSL draws information from richer contextual clues provided by other modalities. Furthermore, multimodal approaches benefit from transfer learning across different domains. For example, it can use semantic descriptions in text to guide the visual classification of novel images. Thus, it compensates to some extent the lack of single-modal data in FSL.

For instance, **Contrastive Language–Image Pretraining (CLIP)** [Radford et al., 2021] learns powerful representations by aligning image and text embeddings in a shared space. This is great for FSL as this enables classification of unseen images using only their textual descriptions or class names. Even without visual examples, the model can infer what it might look like through linguistic context alone.

However, there are limitations. Multimodal models require large and curated datasets with carefully aligned modalities (e.g. each image with its proper description), which can be hard to obtain. Additionally, the alignment can be noisy, it may be ambiguous, incomplete, or inconsistent with the visual content. For example, a caption like “a person doing something interesting” provides little concrete information for classification.

In this work, we will try to use multimodal learning to improve the classification of images with the aid of some class descriptions obtained using an LLM.

2.2 Large Language Models

Large Language Models (LLMs) are large generative deep neural networks designed to process and generate human language. Their defining characteristics are their scale, being composed of billions or even trillions of parameters, and their pretraining strategy, which allows them to understand language and basic world knowledge. Famous examples of LLMs include the GPT (Generative Pretrained Transformer) by OpenAI, Google’s Gemini, Meta’s LLaMA, DeepSeek...

Technically, modern LLMs are built upon the transformer architecture, introduced by [Vaswani et al., 2017]. In particular, LLMs use an autoregressive variant of the transformer, where the model generates text by predicting one token at a time based on previous generated tokens. This structure allows them to be trained as generative models that can produce coherent, contextually appropriate text.

LLMs are pre-trained on vast amounts of data using self-supervised learning, a form of unsupervised learning where the ground truth is derived directly from the input data itself. The model learns by solving proxy tasks that require understanding the structure and meaning of language. Common pretraining objectives include masked language modeling, i.e. predicting missing words in a sentence (see the example in Figure 2.6), and sentence order prediction, where the model must determine the order of the sentences in a paragraph. This procedure grants them the ability to understand grammar and languages and provides lots of knowledge about the world.

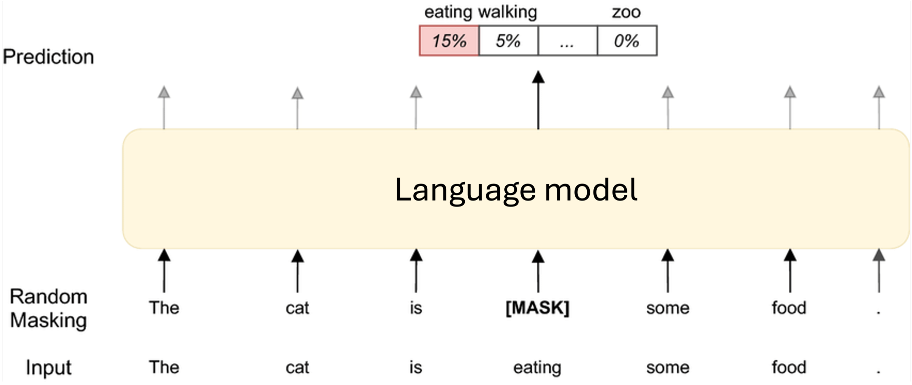


FIGURE 2.6: Example of masked language modeling. One word is hidden in the sentence and the Language model has to guess it based on the sorrounding words.

Later, LLMs are typically fine-tuned with supervised learning on curated, task-specific datasets. This step helps the model follow human prompts more reliably and safely and allows them to perform on real-world applications.

In recent years, LLMs have also expanded into multimodality, moving beyond text to incorporate other data types such as images, audio, and video. Thanks to their vast world knowledge and powerful pattern recognition abilities, multimodal LLMs can interpret images with natural language prompts, answer questions about visual content, generate image captions, ...

2.2.1 In-Context Learning

LLMs are great for any Natural Language Processing (NLP) tasks, such as translation, summarization, question answering, and reasoning. But they also accumulate great knowledge about the world, so they can apply it to solve new tasks. Fun fact is that this ability was not programmed directly, it emerged as models got bigger and were trained on more data.

In-Context Learning (ICL) is a key feature of large language models where they "learn" new tasks without changing their internal weights. Instead of retraining, the model is conditioned by a few examples from the prompt and it is able to generalize the pattern purely from context. For example, as we will see later in this thesis, this mechanism enables zero-shot, one-shot, and few-shot image classification without being never been explicitly trained on them.

Compared to traditional meta-learning, ICL is much more fast, flexible, and easy to use, as it does not require any parameter updates or training. However, it is not very reliable as it is sensitive to prompt design, it does not retain memory between uses, and can fail on tasks that require deep reasoning or structured adaptation.

2.2.2 Prompt Engineering

As previously discussed, prompts are fundamental to how LLMs operate. They define both the communication interface and the specific function the model is expected to perform. Therefore, the branch of Prompt Engineering has gained a lot of popularity recently, becoming a very fundamental part of any interaction with LLMs.

Prompt engineering refers to the process of designing and optimizing input prompts to guide the behavior of LLMs, since it defines the ICL tasks.It is different for each model and takes some time to adjust.

Techniques such as chain-of-thought prompting [Wei et al., 2022] have also shown that guiding the model to reason step-by-step can significantly improve performance on complex tasks like arithmetic or logic. More recently, methods like "automatic prompt generation" [Zhou et al., 2022] or "instruction tuning" [Wang et al., 2022] have enabled LLMs to generalize better across unseen tasks. As models become more powerful but also more opaque, prompt engineering serves as an interface for control and interpretability.

2.3 Ensembles in Machine learning

Ensemble methods are a very powerful strategy in machine learning. It consists on improving the model performance by combining the outputs of multiple models to produce a single, often more robust and accurate prediction. The core idea is that models by themselves may make errors in different ways, but aggregating diverse "opinions" (e.g., different architectures, training strategies, or modalities), the ensemble can correct for individual weaknesses and take advantage of complementary strengths.

Ensemble strategies can be categorized into several types:

1. **Bagging** (Bootstrap Aggregating) [Breiman, 1996]: Models are trained independently on random subsets with replacement (bootstrapping) of the data, and their outputs are averaged or voted upon. This approach reduces variance without increasing bias, making it effective for models sensitive to data changes.

   One classical example of bagging is the **Random Forest** [Breiman, 2001], where multiple decision trees are trained on random subsets of the data and features. The final prediction is typically made through majority voting (for classification) or averaging (for regression). Despite their simplicity, Random Forests remain a strong baseline in many tasks due to their robustness and interpretability.

2. **Boosting**: Models are trained sequentially, where each new model focuses on correcting the mistakes of its predecessors. Well-known examples are AdaBoost [Freund et al., 1997], and Gradient Boosting Machines like XGBoost [Chen et al., 2016].

3. **Stacking** Wolpert, 1992: Multiple models are trained on the same data and their output is combined by a meta-learner trained to learn the optimal way to integrate them. Unlike bagging or boosting, stacking typically uses different types of algorithms trained on the same dataset, and the meta-learner learns how to best combine their outputs.

In this work, we focus on meta-learning approaches, specifically metric-based methods, which are the ones that are having most success nowadays. In particular, we build upon Prototype Networks using the mean, a very simple but effective approach to classify images. Also, we take on some multimodal learning, as we incorporate another modality data in the form of text class descriptions.

Moreover, we will use LLMs to generate the definitions, we use an LLM. In this case, prompt engineering serves as a crucial tool to refine the semantic representations.

Finally, we use both the only-visual approach and the semantic-aided approach at the same time, combining their different inductive biases to generate better predictions in a stacking ensemble approach.

Chapter 3

Methodology

In this section, we present our model for few-shot classification. Our approach builds upon the SemFew framework [Zhang et al., 2024]. Our primary contributions focus on enhancing the semantic alignment module to improve overall performance.

We present the overall structure of the model in Figure 3.1 and our proposal for the semantic module in Figure 3.2.

The model is quite simple. During training, text descriptions and class images are encoded, concatenated, and passed through the Alignment module, which is trained to minimize the distance between each sample and the class prototypes. During testing, the frozen modules combine semantic features and text encodings to form class prototypes, which are then used to classify queries based on similarity to the query image encodings.

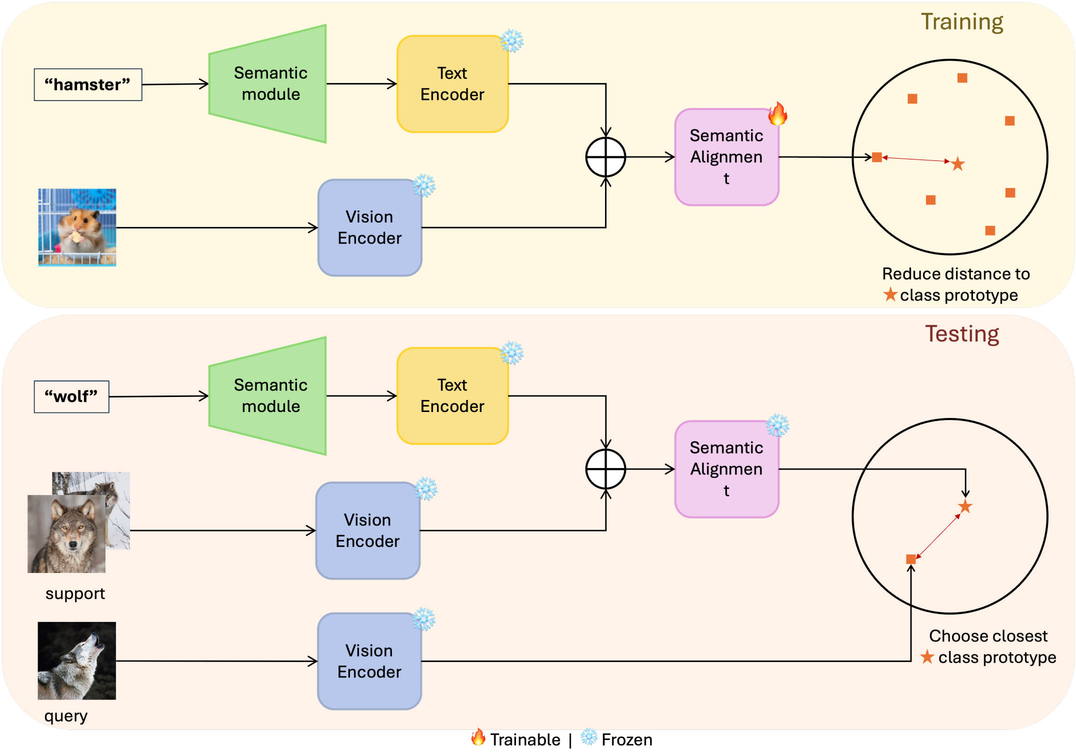


FIGURE 3.1: Diagram of the training and testing stages of the model.

Our method follows a metric-based meta-learning approach, specifically using the Prototypical Network paradigm. We compute a prototype for each class by fusing the data provided by the support samples and the semantic alignment. Then,

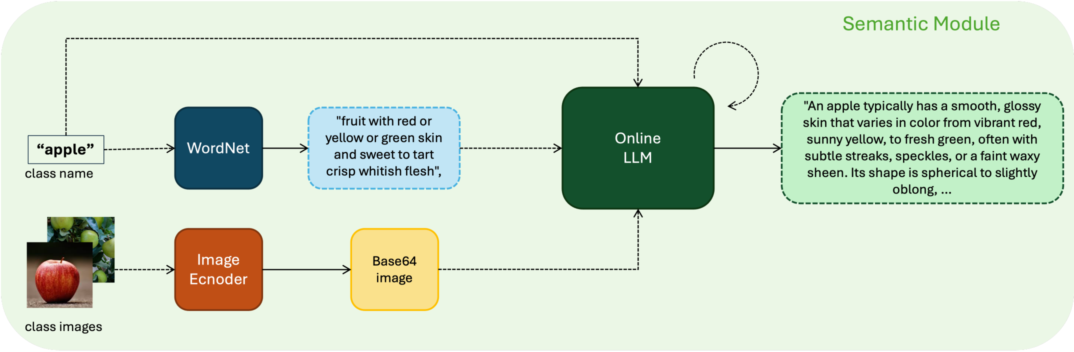

Semantic Module
"apple"
class name
WordNet
"fruit with red or yellow or green skin and sweet to tart crisp whitish flesh",
Online LLM
"An apple typically has a smooth, glossy skin that varies in color from vibrant red, sunny yellow, to fresh green, often with subtle streaks, speckles, or a faint waxy sheen. Its shape is spherical to slightly oblong, ...
class images
Image Ecnoder
Base64 image

FIGURE 3.2: Diagram for the Semantics module. The class name, its WordNet description and images from the class are possible inputs for the Online LLM, who will generate the final description of the class.

for each query sample, we predict its class by measuring the distance between its embedding and each class prototype.

Additionally, our approach falls within the Multimodal Learning category, as it integrates information from both visual and textual modalities. Furthermore, it can be viewed as a stacking ensemble, because we combine two different approaches to classification and learn the proper way to combine them.

3.1 Visual-Only Approach

Few-shot learning datasets rely on two disjoint datasets: a base set $D_{\text{base}} = \{(x, y) \mid x \in X_{\text{base}}, y \in C_{\text{base}}\}$ used for learning a feature space, and a novel set $D_{\text{novel}} = \{(x, y) \mid x \in X_{\text{novel}}, y \in C_{\text{novel}}\}$ used for evaluation, where x is the image, y is the class label and C is the label set. It is important to note that the label sets are disjoint: $C_{\text{base}} \cap C_{\text{novel}} = \emptyset$.

During evaluation, an N-way K-shot task is constructed from the D_{novel}, using a support set $S = \{(x_i, y_i)\}_{i=1}^{N \times K}$ containing N different classes and K examples per class, and a query set $Q = \{(x_i, y_i)\}_{i=1}^{N \times M}$ containing M examples per class. The model has to classify each query image into one of the N classes.

In our approach, we will define a class prototype and classify the query images based on the similarity between its embedding and each class prototype. A simple yet effective strategy to generate the class prototypes, is to use the mean encoding of the samples of the class. Therefore, for class t, its prototype would be denoted as p_t and would be defined as:

$$p_t = \frac{1}{|\mathcal{S}_t|} \sum_{x_i \in \mathcal{S}_t} f(x_i) \tag{3.1}$$

where $\mathcal{S}_t$ is the support set for class t and f is the image encoder function. We will call this visual-only prototype as **proto**. This is our naive approach and it will be improved in the following sections.

Then, for each query image q, its predicted class would be:

$$\hat{y} = \underset{t \in \{1,\ldots,N\}}{\arg\max}\, P(y = t) = \underset{t \in \{1,\ldots,N\}}{\arg\max} \frac{\exp(d(q, p_t))}{\sum_{i=1}^{N} \exp(d(q, p_i))} \tag{3.2}$$

where $d()$ is a similarity function which can be the cosine similarity function:

$$d(a,b) = \frac{a \cdot b}{\|a\| \, \|b\|} \tag{3.3}$$

Or the negative of the euclidean distance:

$$d(a,b) = -\|b-a\|^2 = -\sum_{i=1}^{n} (b_i - a_i)^2 \tag{3.4}$$

where $(a \cdot b)$ is the inner product and $\| \|$ is the norm.

3.2 Semantic Module

The naive approach just mentioned works pretty good but we will improve it using our Semantic Module, shown in Figure 3.2.

As discussed before in this work, some methods tried boosting class knowledge of the model by simply incorporating the encoded class names. However, we can improve it. We propose to transform the class names into short WordNet definitions and expand them into high-quality semantics.

First, we need to handpick the most appropriate WordNet definition for each class. This was necessary because many class names are polysemous, and the default WordNet entry was often inaccurate. Moreover, in some cases, such as "aquarium_fish", which is composed of two words, the term lacked any entry in the WordNet dictionary. We also experimented with using an LLM to automatically select the most accurate definitions, but the results were very lacking. We tried to contact the authors of SemFew for clarification on their WordNet definition selection process, but, unfortunately, we never received any response. Given the fundamental role of the descriptions for the process of generating semantics in our method, we ultimately decided to select the definitions manually, and generate the missing ones.

Once the definitions are ready, we use the LLM to improve them. In this step, we experimented with many different prompting strategies, which will be discussed in the Validation section of the report. Some included incorporating images to the prompt, mentioning the most confusable classes for the class, or even returning the previous output of the same LLM. Our final proposal and the prompt that achieves the best results is:

The following is a definition of the class '{class}': "{definition}". Using this as context (only if useful), write a short visual description to help an AI model recognize '{class}' based on appearance alone. Do not include other classes. One paragraph only.

The resulting descriptions are much better, with rich details truly describing the visual appearance of the classes and their distinguishing features with respect to others.

3.3 Semantic Alignment

After obtaining the high-quality semantics in the step before, now we need to incorporate them into the model so that it can use the information to make better class prototypes. The Semantic Alignment module aims to enhance class prototypes by aligning multimodal information (visual and semantic) to generate the class cluster centers.

As shown in Figure 3.1, the image encoding and the semantic are concatenated into one long feature vector which is fed to the Alignment module. This module will fuse both sources of data to generate an improved prototype.

Contrary to intuition, we deliberately chose a minimal design for the module. In the few-shot learning paradigm, there is very limited data, and a complex model would risk overfitting. We kept the architecture as simple as possible, a two-layer MLP. The design of the Semantic Alignment module is illustrated in Figure 3.3.

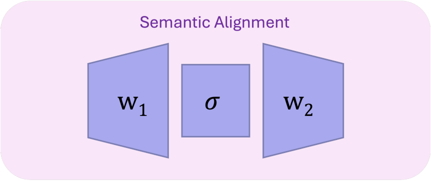


FIGURE 3.3: Diagram of the Semantic Alignment module. It is composed by two linear layers with weights W_1 and W_2 and an activation function σ in between.

As shown in Figure 3.1, this module is the only one that we train, the rest are pre-trained and frozen.

During training, we encode every image and its corresponding class description and concatenate them. The resulting representations are fed to the Alignment module to make a prototype proposal. The module is updated to minimize the distance between each proposal and the center of each class, generated using the mean of all available images for the class. The objective function to optimize is the following.

Let V denote the visual features, T the text features and C the class centers. The Semantic Alignment module learns an alignment between $V \oplus T$ and C using the following loss function:

$$\min_{W_1, W_2} \mathcal{L}\left(\sigma((V \oplus T)^\top W_1)W_2, C\right), \tag{3.5}$$

where W_1 and W_2 are learnable parameters and σ is an activation function.

During testing, we use a N-way K-shot set that is randomly sampled from the novel set. For each image in the support set, we encode it and its class description using the visual encoder and the text encoder respectively. The result is concatenated and passed through the Semantic Alignment module, now frozen. This represents the feature vector for each support sample. The class prototype is the mean of the feature vectors for all K support samples of a class. We will call it the generated prototype or **gen_proto**, or r_t in equation (3.6):

$$r_t = \frac{1}{K}\sum_{i=1}^{K} h(V_i \oplus T_t) \tag{3.6}$$

where V_i are the visual features of the image, T_t are the text features of the class, and $h()$ is the Semantic Alignment function.

This method already outperforms the naive prototype approach mentioned in the introduction of this section, where the class prototypes are computed using only the visual features:

$$u_t = \frac{1}{K}\sum_{i=1}^{K} V_{t,i} \tag{3.7}$$

However, to further improve the performance of our approach, we combine the two approaches together. This helps to keep the results more grounded in the visual aspect and acts as a residual layer, ensuring that we do not lose any relevant visual information. The merging is performed using convex combination regulated by a k factor:

$$p_t = kr_t + (1 - k)u_t, \tag{3.8}$$

where $k \in [0, 1]$ is the fusion factor, r_t is the generated prototype and u_t is the naive prototype.

We considered traning k as a new learnable parameter of the model, but the results showed that its performance was stable over a wide range of values. Therefore, we decided to select the best performing value based on the training results.

Once the refined prototypes are obtained, we classify each query image by identifying the class whose prototype is closest to the image's encoded representation. This is done using the same prediction formula as in the naive approach (Equation (3.2)), using the cosine distance as the similarity metric.

Chapter 4

Validation

In this section, we present the results of our experiments. We begin by stating the experimental setup, followed by an evaluation of our final proposal and its performance compared to state-of-the-art methods. Finally, we present a detailed ablation study to justify the key design decisions, with special emphasis on the prompt selection.

4.1 Experimental Setup

In the following, we describe the specific conditions under which our experiments were conducted and their specific implementation details.

Our work builds upon the work from SemFew framework [Zhang et al., 2024], which was the foundation for both our codebase and evaluation protocol. It was our baseline to surpass. In most of the testing, we recreated and evaluated their code and ours under the same conditions to ensure a fair comparison and isolate the impact of our proposed changes to the Semantic Module. We recreated their performance on our machine and evaluated both results using the same metrics.

4.1.1 Datasets

Dataset	Image Size	Split	Description
CIFAR-FS	32×32 RGB	64 train, 16 val, 20 test	Simple, low-resolution dataset with high inter-class similarity. Widely used in our experiments.
FC100	32×32 RGB	60 train, 20 val, 20 test	More robust split based on superclasses (12/4/4 split). Reduces information leakage across splits by grouping semantically related classes together. More challenging than CIFAR-FS.
MiniImageNet	84×84 RGB	64 train, 16 val, 20 test	Higher resolution and greater visual diversity. More complex and realistic for real-world scenarios.

TABLE 4.1: Table describing tje datasets used.

We evaluate the proposed method on three established FSL datasets: **CIFAR-FS** [Bertinetto et al., 2019] and **FC100** [Oreshkin et al., 2018] from the CIFAR-100 dataset [Krizhevsky, 2009] and **MiniImageNet** [Vinyals et al., 2016] from ImageNet [Deng et al., 2009]. Each dataset has its classes already split into train/val/test to evaluate few-shot learning capabilities. Each class has 600 images. More information about the datasets in Table 4.1.

4.1.2 Implementation details

Architecture

Our model is composed of three trainable components: the vision encoder, the text encoder, and the semantic module.

- **Vision Encoder.** The majority of our experiments were done using a pre-trained ResNet-12 backbone [He et al., 2016], due to its good results and fast computational speed, which allowed us to evaluate results faster. To validate our approach, we also evaluated performance with Swin-T [Liu et al., 2021], which is another typical benchmark, and CLIP [Radford et al., 2021]. The output dimension for each encoder is: 640 for ResNet-12, 768 for Swin-T, and 512 for CLIP.
- **Text Encoder.** In this case, we focused on the CLIP pre-trained text encoder. Prior works also considered using BERT [Devlin et al., 2019], however, there are many insights stating that CLIP outperforms BERT in this context [Zhang et al., 2024; Yan et al., 2022; Chen et al., 2023c], so we decided to focus on it. For both the CLIP text and vision encoders, we used the open-source implementation from OpenCLIP [Cherti et al., 2022], which outputs 512-dimensional embeddings.
- **Semantic Alignment Module.** To combine visual and textual embeddings, we use a very lightweight Semantic Alignment module consisting of two fully connected layers with a LeakyReLU activation function [Maas et al., 2013] in between. The hidden layer's dimension is 4096 and the output dimension is the same as the visual embedding.

Semantics module

The semantics module, detailed in Figure 3.2, supports many different usage protocols to generate the text descriptions. The semantics can be extracted using the class names, randomly selected images of the class or WordNet descriptions. The input(s) chosen will be fed to the LLM with the proper prompt to generate the detailed description. Some of the protocols also allow to resend the output text to the LLM to refine it with different criteria.

We experimented with different state-of-the-art LLMs. There is summary of their characteristics in Table 4.2.

The LLMs where chosen to be varied, have state-of-the-art performance and be free (or at least inexpensive). It was also valued that they had good results and low latencies in their free version. Multimodal input for both text and image was also important, as it was required for many of our tests. To access them, we used the OpenAI, GenAI or Groq APIs.

Training Protocol

During training, we freeze the weights of the pre-trained encoders (both visual and text) and train only the alignment module.

Model	Origin	Params	Description
GPT-4	OpenAI, 2023	≈1.76T	Strong reasoning and instruction-following capabilities; most famous model. GPT-3.5 was used in SemFew.
LLaMA-3	Meta, 2024	8B	Very low latency, especially in Groq API; widely used in our experiments.
LLaMA-4 Scout	Meta, 2025	17B active, 109B total	Mixture-of-Experts architecture with 16 experts; very limited free access.
LLaMA-4 Maverick	Meta, 2025	17B active, 400B total	Mixture-of-Experts with 128 experts; very limited free access.
Gemini-2 Flash Lite	Google, 2024a	N/A	Faster, simplified version of Gemini 2; famous for multimodal capabilities (text, image, audio, video).
Mistral Small 3.1	MistralAI, 2025	24B	Compact and efficient.
Gemma-3	Google, 2024b	27B	Open-source model; we used the largest available version.
Qwen-2.5	Alibaba, 2024	72B	Open-source model, particularly good at recognizing common objects in images.
DeepSeek R1	DeepSeek-AI, 2025	37B active, 671B total	Open-source model using Mixture-of-Experts; great detail.

TABLE 4.2: Summary of LLMs used in this work.

All experiments were run on Kaggle Notebooks using an NVIDIA P100 GPU. We use a batch size of 128 and the Adam optimizer [Kingma et al., 2015] with a learning rate of 1×10^{-4}, which decays after 30 epochs by a factor of 0.1. The model is trained for a maximum of 50 epochs. We save the best-performing model on the validation set and use it for final testing.

Evaluation Protocol

We use the standard N-way K-shot few-shot learning settings, using 600 randomly sampled tasks. We focused on *1-shot* and *5-shot* and we used 15 query samples per class.

For each experiment, we report the mean accuracy through all tasks and the 95% confidence interval.

4.2 Few-Shot Results

Table 4.3 summarizes the performance of our approach compared to other state-of-the-art methods on CIFAR-FS, which is our primary evaluation dataset. Our main goal for this work was to outperform our baseline, which is inspired by the SemFew framework[1], and so we did, even achieving state-of-the-art results for the Few-shot classification task on CIFAR-FS, improving on the main benchmarks.

[1] It is worth noting that the accuracy values reported for SemFew in the original paper and the baseline results obtained empirically do not exactly match. In some cases, such as with CIFAR-FS, the results were quite close, likely due to minor differences such as random seed differences during testing. However, for more complex datasets like FC100 and MiniImageNet, the discrepancies were larger and unlikely to be caused by such small variations alone. Therefore, in our results report, we call "Baseline" to the reproduced results using their official code and embeddings.

	Method	Author	Backbone	CIFAR-FS	
				1-shot	**5-shot**
V	ProtoNet	Snell et al., 2017	ResNet-12	72.20 ± 0.70	83.50 ± 0.50
V	MetaOptNet	Lee et al., 2019	ResNet-12	72.80 ± 0.70	84.30 ± 0.50
V	MABAS	Kim et al., 2020	ResNet-12	73.51 ± 0.92	85.65 ± 0.65
V	RFS	Tian et al., 2020	ResNet-12	71.50 ± 0.80	86.00 ± 0.50
V	SUN	Dong et al., 2022a	ViT-S	78.37 ± 0.46	88.84 ± 0.32
V	FewTURE	Hiller et al., 2022	Swin-T	77.76 ± 0.81	88.90 ± 0.59
V	CPEA	Hao et al., 2023	ViT-S/16	77.82 ± 0.66	88.98 ± 0.45
V	SMKD	He et al., 2022	ViT-S	80.08 ± 0.18	**90.63 ± 0.13**
S	SP-CLIP	Chen et al., 2023b	Visformer-T	82.18 ± 0.40	88.24 ± 0.32
S	SemFew-Res	Zhang et al., 2024	ResNet-12	83.65 ± 0.70	87.66 ± 0.60
S	SemFew-Trans	Zhang et al., 2024	Swin-T	84.34 ± 0.67	89.11 ± 0.54
S	FewVS-Res	Li et al., 2024	ResNet-12	84.40 ± 0.35	88.14 ± 0.31
S	Baseline	Ours	ResNet-12	82.72 ± 0.71	86.98 ± 0.56
S	Ours	Ours	ResNet-12	83.51 ± 0.70	87.08 ± 0.63
S	Baseline	Ours	Swin-T	84.32 ± 0.68	88.70 ± 0.62
S	Ours	Ours	Swin-T	**84.91 ± 0.69**	89.36 ± 0.59

TABLE 4.3: Comparison with state-of-the-art methods on CIFAR-FS under 5-way 1-shot and 5-way 5-shot settings. Visual methods are marked with V, semantic methods with S.

Moreover, in Tables 4.4 and 4.5, we also report our accuracy results compared to many state-of-the-art methods, both from visual-only approaches and semantic-aided approaches. From these results, we can see that our approach consistently improves the baseline, which was our objective. It also provides a competitive point of view, able to compete with other much more computationally expensive methods.

Overall, our approach shows the best results in the 1-shot scenario, proving the semantic aid to be key for low-data problems. In the 5-shot scenario however, while it provides a substantial performance boost, purely visual approaches with strong representation learning capabilities, such as SMKD and CPEA, continue to achieve superior results, as they use a much stronger visual backbone.

4.3 Ablation study

In this section, we analyze the contribution of each component in our approach to validate each design decision.

4.3.1 Alignment Source

First of all, we need to analyze how much each module contributes to the model's final performance, to see if it is even worth it contemplating the semantic module. As shown in Table 4.6, the semantic component plays a fundamental role. In the 5-way 1-shot scenarios, the semantic approach clearly outperforms the visual one, achieving results up to 9 points higher and significantly driving the overall performance. In the 5-way 5-shot cases, the visual component becomes more important, having better results than the semantic one. However, neither component alone is

				FC100	
	Method	**Author**	**Backbone**	**1-shot**	**5-shot**
V	ProtoNet	Snell et al., 2017	ResNet-12	41.54 ± 0.76	57.08 ± 0.76
V	TADAM	Oreshkin et al., 2018	ResNet-12	40.10 ± 0.40	56.10 ± 0.40
V	MetaOptNet	Lee et al., 2019	ResNet-12	47.20 ± 0.60	55.50 ± 0.60
V	MABAS	Kim et al., 2020	ResNet-12	42.31 ± 0.75	58.16 ± 0.78
V	RFS	Tian et al., 2020	ResNet-12	42.60 ± 0.70	59.10 ± 0.60
V	FewTURE	Hiller et al., 2022	Swin-T	47.68 ± 0.78	63.81 ± 0.75
V	CPEA	Hao et al., 2023	ViT-S/16	47.24 ± 0.58	65.02 ± 0.60
V	SMKD	He et al., 2022	ViT-S	50.38 ± 0.16	**68.37 ± 0.16**
S	SP-CLIP	Chen et al., 2023b	Visformer-T	48.53 ± 0.38	61.55 ± 0.41
S	Baseline	Ours	ResNet-12	51.16 ± 0.74	60.48 ± 0.76
S	Ours	Ours	ResNet-12	52.62 ± 0.71	60.85 ± 0.74
S	Baseline	Ours	Swin-T	50.89 ± 0.80	62.59 ± 0.78
S	Ours	Ours	Swin-T	**52.68 ± 0.79**	62.76 ± 0.78

Table 4.4: Comparison with state-of-the-art methods on FC100 under 5-way 1-shot and 5-way 5-shot settings.

				MiniImageNet	
	Method	**Author**	**Backbone**	**1-shot**	**5-shot**
V	MatchNet	Vinyals et al., 2016	ResNet-12	65.64 ± 0.20	78.72 ± 0.15
V	ProtoNet	Snell et al., 2017	ResNet-12	62.39 ± 0.21	80.53 ± 0.14
V	TADAM	Oreshkin et al., 2018	ResNet-12	58.50 ± 0.30	76.70 ± 0.30
V	CAN	Hou et al., 2019	ResNet-12	63.85 ± 0.48	79.44 ± 0.34
V	CTM	Li et al., 2019	ResNet-18	64.12 ± 0.82	80.51 ± 0.13
V	RFS	Tian et al., 2020	ResNet-12	62.02 ± 0.63	79.64 ± 0.44
V	FEAT	Ye et al., 2020	ResNet-12	66.78 ± 0.20	82.05 ± 0.14
V	Meta-Baseline	Chen et al., 2021	ResNet-12	63.17 ± 0.23	79.26 ± 0.17
V	SUN	Dong et al., 2022a	ViT-S	67.80 ± 0.45	83.25 ± 0.30
V	FewTURE	Hiller et al., 2022	Swin-T	72.40 ± 0.78	86.38 ± 0.49
V	FGFL	Cheng et al., 2023	ResNet-12	69.14 ± 0.80	86.01 ± 0.62
V	Meta-AdaM	Gao et al., 2023	ResNet-12	59.89 ± 0.49	77.92 ± 0.43
V	CPEA	Hao et al., 2023	ViT-S/16	71.97 ± 0.65	**87.06 ± 0.38**
S	KTN	Peng et al., 2019	Conv-128	64.42 ± 0.72	74.16 ± 0.56
S	AM3	Xing et al., 2019	ResNet-12	65.30 ± 0.49	78.10 ± 0.36
S	TRAML	Xu et al., 2022	ResNet-12	67.10 ± 0.52	79.54 ± 0.60
S	AM3-BERT	Yan et al., 2021	ResNet-12	68.42 ± 0.51	81.29 ± 0.59
S	SP-CLIP	Chen et al., 2023c	Visformer-T	72.31 ± 0.40	83.42 ± 0.30
S	Baseline	Ours	ResNet-12	72.91 ± 0.65	78.92 ± 0.78
S	Ours	Ours	ResNet-12	73.21 ± 0.66	79.77 ± 0.78
S	Baseline	Ours	Swin-T	74.85 ± 0.70	81.74 ± 0.75
S	Ours	Ours	Swin-T	**75.02 ± 0.69**	82.72 ± 0.79

Table 4.5: Comparison with state-of-the-art methods on MiniImageNet under 5-way 1-shot and 5-way 5-shot settings.

	CIFAR-FS		MiniImageNet	
Method	**5-way 1-shot**	**5-way 5-shot**	**5-way 1-shot**	**5-way 5-shot**
V	72.55 ± 0.92	86.07 ± 0.65	59.87 ± 0.80	77.73 ± 0.88
S	80.57 ± 0.75	81.71 ± 0.74	68.52 ± 0.69	68.27 ± 0.96
V + S	**83.51 ± 0.70**	**87.08 ± 0.63**	**73.21 ± 0.66**	**79.77 ± 0.78**

TABLE 4.6: Results on different alignment sources. V means visual features, S represents semantic features, V + S both.

sufficient to compete with the state of the art. The true potential lies in combining both modalities, complementing each other.

4.3.2 Prompt

A large part of this work involved determining the best way to interact with the chosen LLM in order to obtain the most precise and informative descriptions i.e. prompt engineering.

It is important to highlight that a well-crafted prompt is fundamental to the success of the model. As shown in Figure 4.1 and Table 4.7.

Method	Shot	V Acc (%)	S Acc (%)	k	Mix Acc (%)	Gap
Only Names	1-shot	72.55 ± 0.92	72.20 ± 0.80	0.22	75.06 ± 1.03	2.51
WordNet	1-shot	72.55 ± 0.92	71.33 ± 0.99	0.29	73.88 ± 1.03	1.33
LLM help	1-shot	72.55 ± 0.92	80.57 ± 0.75	0.20	**83.51 ± 0.70**	**10.96**
Only Names	5-shot	85.82 ± 0.89	75.31 ± 1.19	0.58	86.60 ± 0.87	0.78
WordNet	5-shot	86.07 ± 0.65	73.54 ± 1.17	0.60	86.43 ± 1.03	0.36
LLM help	5-shot	86.07 ± 0.65	81.71+0.74	0.54	**87.08+0.63**	**1.01**

TABLE 4.7: Comparison of methods with different text embeddings and their corresponding accuracies and metrics. V accuracy for the only visual part, S accurcy for the semantic part.

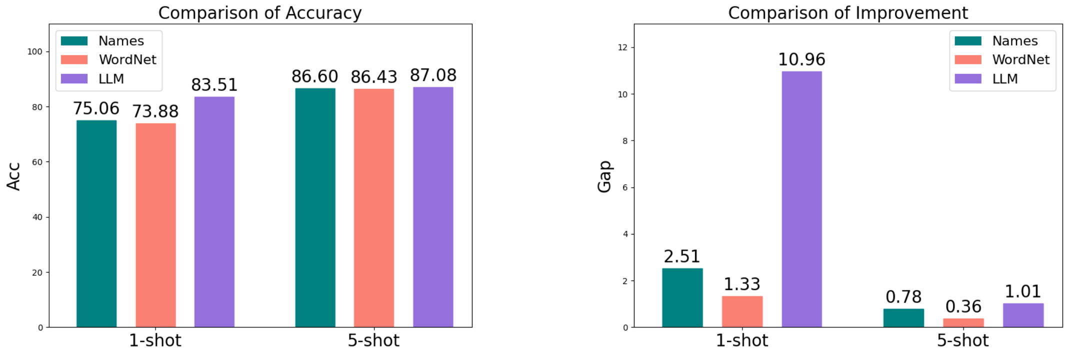


FIGURE 4.1: Results using just names, just WordNet descriptions or the full LLM proposed descriptions for both 1-shot and 5-shot. The first table shows the accuracy results and the second the gap between the final accuracy and the only-visual accuracy.

Just using the class names or the WordNet descriptions is not enough. Performance improves significantly with a good prompt. Moreover, we analyzed the distribution of the classes and how they cluster, as illustrated in Figures A.1, A.2 and A.3 in the Annex A. From this analysis, we conclude that using only the class names does not provide sufficient information. While WordNet descriptions offer more detail that helps to better separate classes, much of this information is not relevant for the visual classification tasks. Therefore, it is necessary to refine these descriptions through prompts specifically tailored for the LLM.

Our base prompt, proposed in SemFew [Zhang et al., 2024], asks the LLM to refine the WordNet description by adding visual characteristics. We will refer to this as Prompt 0. Upon examining this prompt, we identified certain shortcomings: it relied too heavily on the original WordNet text, often resulting in descriptions with many irrelevant details for visual classification. Additionally, we hypothesized that it would be useful to mention potentially confusing classes to help the model focus on the most discriminative features.

Despite some differences, all proposed prompts shared some key similarities: they requested visual information specific to each class (given our goal of visual classification) and constrained the output to a single paragraph long. We further observed that longer descriptions generally led to better performance, up to an equilibrium point around five lines, beyond which additional information had minimal impact.

In the following, we present the various ideas we proposed for improving the descriptions, along with their advantages and disadvantages. We then provide the exact prompts used to evaluate each approach. Finally, we present a study of the resulting descriptions from the calls using said prompts for an example class: "baby", followed by both qualitative and quantitative analyses of these outputs.

The improvement ideas we explored were:

- **Remove WordNet descriptions**. WordNet definitions often include multiple meanings and, in some cases, are missing entirely, like in the of the class "aquarium_fish". For example, take the definitions provided for the class "baby":

  - '(slang) sometimes used as a term of address for attractive young women',
  - 'a very young child (birth to 1 year) who has not yet begun to walk or talk',
  - 'the youngest member of a group (not necessarily young)',
  - 'an immature childish person', 'an unborn child; a human fetus',
  - 'a very young mammal',
  - 'a project of personal concern to someone',
  - 'treat with excessive indulgence'

  Only one of these definitions is acceptable for the images in our dataset, and it is not even the first.

  At first, we considered removing them from the prompt altogether, relying solely on the LLM-generated descriptions. However, this sometimes introduced more uncertainty, as polysemous words could lead to misunderstandings. For this reason, we decided to manually curate the set of WordNet definitions, carefully selecting or completing them when missing. These manually selected descriptions will be used in the prompts that required them, additional context to help avoiding ambiguities.

- **Guide the prompt toward specific visual aspects**. For example, we explicitly asked the LLM to focus on "color, shape, texture, size, or key parts" of each class. While this approach did help emphasize visual features, it often led to overly detailed descriptions or too specific details that could not actually be seen in the images.

- **Mention all other classes to highlight discriminative features**. The idea here was to explicitly mention the other classes so the LLM would emphasize distinguishing characteristics. However, since there are so many classes, the LLM rarely followed this instruction successfully.

- **Mention only the most commonly confused classes.** We analyzed the confusion matrices from previous results and applied K-means clustering to identify the classes most similar to the target class. For example, for "baby", the most confused classes were "man", "woman", "boy" and "girl." In this case, the LLM did correctly address the request and provided relevant distinguishing details. However, it often structured the text by directly comparing to these other classes (e.g., "The difference between a boy and a girl is..."). Simply mentioning other class names can distort the embedding in undesirable ways, even if they are mentioned only for comparison. We validated this effect through the analysis shown in Figure A.4.

- **Explicitly request not to mention other classes.** Following the findings above, we tried to explicitly instruct the LLM not to mention other classes. Unfortunately, many current LLMs struggle to follow negative instructions, and this request was often ignored. More details on this limitation are discussed in the section on LLM Validation.

- **Include class images in the prompt.** If the goal is to describe an image class accurately, what better way than providing actual images? The idea was to follow the 1-shot or 5-shot setup: in the 5-shot case, for example, we would give the support set images to the LLM so it could generate a description using its vision capabilities. Due to limited access to LLMs, we could not make as many calls as required for every episode, so we tested this by using five random images per class from the dataset.

  The results were quite good, in fact, among the most visually accurate. However, in many cases, the descriptions included references to irrelevant details about the form of the images, such as "blurry definition" or mentions of non discriminative elements in the images, which sometimes worsen the recognition performance. This issue was more pronounced in the 1-shot case. We also have to consider the possibility that some of this information would be redundant as it could be easily extracted from the vision module.

- **Emphasize visual classification from an AI.** By explicitly stating in the prompt that the text would be used for image classification by an AI model, we can guide the LLM to focus more on visual aspects typically detectable by computer vision systems.

- **Multi-prompt strategy.** Finally, we developed a multi-stage prompt strategy. The first stage would use the WordNet description and the most confusing classes to request additional detailed information of all types without length constraints. The second stage would incorporate images of the class and ask the LLM to summarize the content, focusing on visual features present in the provided examples. The third stage would refine the result, removing any mention of other classes while preserving the semantic content.

The main limitation of this approach is that not all models support image inputs in prompts. Thus, despite its great potential, we were unable to fully exploit this method.

In the Appendix B, we provide a detailed list of the different prompts that were tested, along with examples of their outputs for the class "baby." Each prompt is accompanied by a qualitative commentary analyzing why that particular approach worked or did not work. Additionally, Table 4.8 presents a quantitative analysis of the prompt results.

Prompt	**Entropy**	**Min D.**	**Mean D.**	**Acc.**	**Most Confused Classes**
Default	4.9247	0.0202	0.1453	82.72 ± 0.71	girl ↔ woman dolphin ↔ whale snake ↔ lizard
Just names	**4.9771**	0.0958	0.3428	75.06 ± 1.03	man ↔ plain boy ↔ girl boy ↔ man
WordNet	4.7199	0.0593	0.4226	73.88 ± 1.03	man ↔ woman tiger ↔ leopard tulip ↔ poppy
Prompt 0	4.8757	0.1180	0.3595	82.44 ± 0.72	couch ↔ bed boy ↔ girl chair ↔ table
Prompt 1	4.9753	0.1073	0.3586	81.44 ± 0.74	girl ↔ woman boy ↔ man bear ↔ rabbit
Prompt 2	4.7490	0.0924	0.3423	**83.45 ± 0.71**	girl ↔ woman man ↔ woman couch ↔ bed
Prompt 3	4.8819	0.1084	0.3719	81.70 ± 0.70	girl ↔ woman bear ↔ rabbit man ↔ woman
Prompt 4	4.8887	0.1300	**0.3947**	82.84 ± 0.74	lion ↔ tiger mouse ↔ squirrel motorcycle ↔ bicycle
Prompt 5	4.8628	**0.1355**	0.3936	81.74 ± 0.69	lobster ↔ crab palm_tree ↔ pine_tree shark ↔ whale

TABLE 4.8: Summary of the performance of the different Prompt approaches. Min D. and Mean D. represent the minimum distance and the mean distance between text embeddings using the cosine distance. The accuracy was evaluated on 5-way 1-shot on CIFAR-FS using the ResNet-12 backbone. The tests were done using LLaMA-3 as it was the model with the fastest latencies.

In conclusion, we found that the prompt providing the best results is:
Prompt 2: Improved knowledge
The following is a definition of the class 'class': "definition".
Using this as context (only if useful), write a short visual description to help an AI model recognize 'class' based on appearance alone. Do not include other classes. One paragraph only.

This prompt effectively combines the clarifying definition from WordNet without being strictly tied to it, using its information only when helpful. It also provides context that guides the LLM to focus on the most visually relevant aspects of the class. Furthermore, this prompt does not require input images, making it compatible with a broader range of LLMs, including text-focused models that do not support image input but can generate exceptionally detailed textual descriptions, as shown in the following section.

4.3.3 LLM

For this project, we considered the following LLMs: GPT-4, LLaMA-3, LLaMA-4 Scout and Maverick, Gemini-2, Gemma-3, Mistral-3.1, Qwen-2.5, and DeepSeek-R1. In this case, conducting a purely qualitative analysis is more challenging. While we could focus on the well-known strengths of each model, for example, GPT's ability to follow detailed instructions, Gemini's conciseness, Qwen's strong image recognition capabilities, or DeepSeek's high level of descriptive detail, these are subjective qualities. We cannot confidently assert that they directly translate into improvements in the resulting embeddings. Therefore, we opted to carry out a quantitative study to objectively evaluate their impact. The results are shown in Table 4.9.

The numbers clearly point to DeepSeek as the winner, and when we look at its descriptions, it's not surprising as they stand out for their remarkable detail and precision. For example, in the description of "apple," it describes its color as: "*vibrant red, sunny yellow, to fresh green, often with subtle streaks, speckles, or a faint waxy sheen*", a level of precision none other has achieved.

Nearly all of the descriptions we used are available in the GitHub repository as JSON files, should anyone wish to explore them further.

4.3.4 Other considerations

Loss formula

During the model improvement process, we considered using a modified loss function to explicitly penalize commonly confused classes, such as "rose" and "poppy." However, this idea was ultimately discarded due to the nature of the few-shot meta-learning paradigm. In this setting, only five classes are compared at a time, and it is rare for specific confusing pairs to appear together in the same episode, which would result in unnecessary complexity in the loss function.

Moreover, each dataset has a different class split: for example, in CIFAR-FS, classes like "boy" and "man" appear in the train and test sets respectively, whereas in FC100, both belong to the test set. None of those classes are in MiniImageNet. As a result, any weight designed to penalize these pairs would often have no practical effect.

It also feels conceptually wrong to introduce such class-specific penalties in a few-shot learning scenario, where the goal is precisely to evaluate the model's ability to adapt to unknown classes.

A different approach could be to penalize embeddings that are overly similar regardless of the class, but this is essentially what we already achieve with our Semantic Alignment strategy.

Prototype generation

We can also vary the way we generate the "centers" or "prototypes." The most common method is to calculate the arithmetic mean, but we also considered alternatives such as clustering algorithms. Several studies suggest that using the mean is

LLM	Entropy	Min D.	Mean D.	Acc.	Most Confused Classes
GPT-4	**4.9481**	0.0743	**0.3957**	79.04 ± 0.79	hamster ↔ mouse hamster ↔ squirrel mouse ↔ squirrel
LLaMA-3	4.7490	0.0924	0.3423	83.45 ± 0.71	girl ↔ woman man ↔ woman couch ↔ bed
Gemini-2	4.6518	0.0747	0.3654	82.84 ± 0.74	mouse ↔ shrew couch ↔ bed girl ↔ woman
LLaMA-4 S	4.8758	0.0977	0.3439	82.50 ± 0.71	bowl ↔ cup man ↔ woman apple ↔ pear
LLaMA-4 M	4.8729	0.0791	0.3349	82.73 ± 0.72	girl ↔ woman boy ↔ man couch ↔ bed
Mistral-3.1	4.7623	0.1051	0.3709	82.74 ± 0.72	motorcycle ↔ bicycle boy ↔ man man ↔ woman
Gemma-3	4.8679	0.0974	0.3704	82.54 ± 0.72	girl ↔ woman motorcycle ↔ bicycle boy ↔ girl
Qwen-2.5	4.7808	0.0829	0.3594	82.86 ± 0.69	girl ↔ woman boy ↔ man couch ↔ bed
DeepSeek-R1	4.8349	**0.1135**	0.3802	**83.51 ± 0.70**	girl ↔ woman man ↔ woman boy ↔ girl

TABLE 4.9: Summary of the performance of the different LLMs. Min D. and Mean D. represent the minimum distance and the mean distance between text embeddings using the cosine distance. The accuracy was evaluated on 5-way 1-shot on CIFAR-FS using the ResNet-12 backbone. The tests were done using their best performing prompt. The GPT-4 results were a bit different. As we did not have access to the API at the time, it was generated with the Chat interface. Probably due to token limitations, the output descriptions were quite short.

good enough for these cases. In experiments with our model, we observed an accuracy of 82.39% using the mean and 82.65% with clustering, a very small difference, especially in few-shot scenarios with very limited support samples, and tending to favor the mean. In this case, we believe the mean is the most efficient and effective approach.

k value calculation

Our approach determines the optimal value of $k \in (0, 1)$ at each validation iteration, using a step size of 0.01. At the end of the execution, we review the performance history for each k and select the one that achieved the best overall results.

We also experimented with introducing a learnable parameter to directly optimize k, but we observed that the results remained very stable and did not differ significantly. As shown in Figure 4.2, the range of optimal values is fairly broad and it is consistently found with the current algorithm. For this reason, we opted to keep the simpler, explicit search approach.

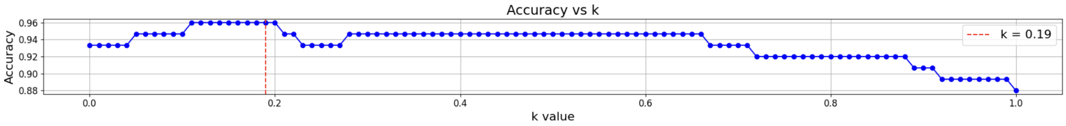


FIGURE 4.2: Accuracy vs k value graphic.

4.3.5 CLIP backbone

In our experiments, we not only tested with ResNet-12 and SWIN-T encoders, we also want to present the results obtained using CLIP in Table 4.10.

Dataset	Encoder	Semantic	k	V Acc (%)	S Acc (%)	Mix Acc (%)
CIFAR-FS	CLIP	Just names	0.57	97.16 ± 0.30	84.04 ± 1.14	97.64 ± 0.27
CIFAR-FS	CLIP	DeepSeek	0.41	97.16 ± 0.30	91.11 ± 0.82	97.72 ± 0.26
FC100	CLIP	DeepSeek	0.56	84.71 ± 0.90	56.29 ± 1.10	85.59 ± 0.87
MiniImageNet	CLIP	DeepSeek	0.46	97.19 ± 0.29	90.13 ± 0.70	97.51 ± 0.24

TABLE 4.10: Summary of results using CLIP encoder for all dataset in 5-way 1-shot classification.

We believed in the potential of CLIP, as it has demonstrated impressive performance on similar tasks. However, the results we obtained exceeded all expectations, achieving near-perfect accuracy on almost every dataset, significantly surpassing any officially published results.

This exceptional performance is likely due to the fact that the pre-trained CLIP model may have been trained on a dataset containing our data. Although it was not trained for our specific task, it may have already seen many of the test images, which would make these results unreliable and not truly a sign of generalization capabilities.

Given this situation, we decided not to include these results in our main performance tables. Instead, we focus on ResNet-12 and SWIN-T, which are the most widely used in the literature and allow for a fair comparison with existing approaches.

4.4 0-shot classification

During the course of this project, we became curious about evaluating the ability of LLMs to classify the images on their own, maybe rendering useless any attempt of doing it any other way. Given their well-known multimodal capabilities and extensive world knowledge, we decided to try it. Additionally, these experiments were useful to decide our choice of LLM to integrate into our final model.

The results in Table 4.11 clearly demonstrate the superior performance of GPT-4. Additionally, these findings further validate that our proposed DeepSeek descriptions are the most effective when used with the CLIP encoder.

However, it is important to note (as mentioned earlier for CLIP) that these results should not be taken at face value. It is highly likely that these models have been exposed to these datasets during their training, which could artificially inflate their performance.

Model	**Accuracy**
CLIP (Just names)	0.8110
CLIP (WordNet descriptions)	0.8420
CLIP (Gemini image descriptions)	0.9005
CLIP (DeepSeek descriptions)	**0.9220**
LLaMA-3	0.8874
Mistral Small 3.1	0.6702
Qwen-2.5	0.8624
Gemma-3	0.8233
Gemini-2.0	0.9211
GPT-4	**0.9639**

TABLE 4.11: Zero-shot classification results using LLMs and CLIP. The tests were done without any support images (true zero-shot setting), but not in a 5-way setup: each image was presented individually and the model was asked to classify it into one of all possible classes. Some additional post-processing was required, as the LLMs did not always strictly follow instructions to return only the class name.

Chapter 5

Conclusions

In this thesis, we have thoroughly explored what few-shot learning and meta-learning are, as well as their many variants and different approaches to addressing data scarcity. We focused on the metric-based approach inspired by the Prototype Networks given their well-demonstrated advantages. In particular, we used the SemFew [Zhang et al., 2024] work as our baseline and aimed to improve it by incorporating a stronger semantic component.

To achieve this, we investigated several enhancement strategies using different data sources, analyzed the most widely used LLMs today, conducted an in-depth prompt engineering study, and examined how these elements interact. We proposed a new prompting method that consistently outperforms the baseline, as validated on multiple benchmarks: CIFAR-FS, FC100, and MiniImageNet.

Furthermore, we also demonstrated how modern LLMs and CLIP models can easily solve the 0-shot problem without any specific training, and we discussed potential directions for the future of few-shot learning.

5.1 Contributions and Limitations

The main contribution of this work is proposing a remarkably simple model to tackle the few-shot learning problem. Excluding the pretrained encoders, the model only requires two trainable fully connected layers in the semantic alignment module to merge the visual and semantic information. Despite its simplicity, it achieves outstanding results, competing with the state of the art, especially in 1-shot tasks, where semantic assistance especially shines. Moreover, apart from demonstrating the usefulness of incorporating semantic guidance in few-shot learning, this thesis also provides a detailed study on the impact of different semantic sources, such as images and large language models, and how they interact with various prompt styles. Such insights may be very useful for future research in this field.

This work has still some limitations when compared to purely visual approaches. While our model performs exceptionally well in 1-shot tasks, the advantage of semantic assistance diminishes as the size of the meta-learning support set increases, and it becomes increasingly important to have a strong visual backbone, an area where we cannot quite compete with other state-of-the-art methods. Additionally, the semantic aid requires knowing the class names beforehand, which is not always guaranteed in real-world scenarios. Our approach also involves the manual selection of WordNet descriptions, future work could focus on automating this process. Finally, it is important to note that our method relies on access to a large language model, which brings its own problems, such as significant water usage and energy consumption, issues that deserve greater public awareness as LLMs become more widespread.

5.2 Future work

In the near future, it is important to update the class descriptions using the latest advances in large language models, as this field is evolving at an incredible pace. Looking further ahead, one could explore new approaches that make even better use of semantic embeddings. From the beginning of this project, we had the idea of using "entities", i.e. instead of relying on full-text descriptions, we could extract distinct visual components and guide the model to recognize these elements, making classification more intuitive and potentially more accurate. This approach would also benefit from the fact that some entities may be shared among the learned and novel classes, simplifying the adaption to new tasks. However, this idea would require a more powerful vision backbone and a deeper integration between feature extraction and entity recognition. Beyond this, it would also be valuable to test these discoveries in less controlled few-shot scenarios to better understand how they perform in real-world conditions.

The field of few-shot learning is incredibly interesting and full of potential that challenges all we know about machine learning. However, my critical opinion is that it is somewhat difficult to get into this field at first because many papers either exaggerate their results or use non-standard methods that deviate from the core principles of meta-learning. It is striking to see how, on platforms like Papers with Code, there are countless works presenting nearly perfect accuracies, while papers published in the principal venues often struggle to exceed 80% accuracy on the same datasets. It also seems to be a common practice to omit or downplay stronger results from other works in order to make new contributions appear more impressive.

Personally, I believe few-shot learning still has a lot to offer, but the current benchmark datasets have become oversaturated. Today, there are multiple ways to achieve almost perfect accuracy, as we demonstrated in the 0-shot classification section, even without additional training. Models like CLIP and LLMs make these tasks almost trivial. At this point, it no longer makes sense to frame progress as simply pushing for higher accuracy scores, overyone can "chase" or train to create a distilled version of CLIP [Li et al., 2024].

The future of this field lies not in chasing higher metrics, but in exploring more realistic and less controlled scenarios to truly understand how these models perform in the real world. We need to re-visit how we think about learning, develop new strategies, and develop new strategies that can genuinely benefit the entire machine learning community.

Appendix A

Extra Figures

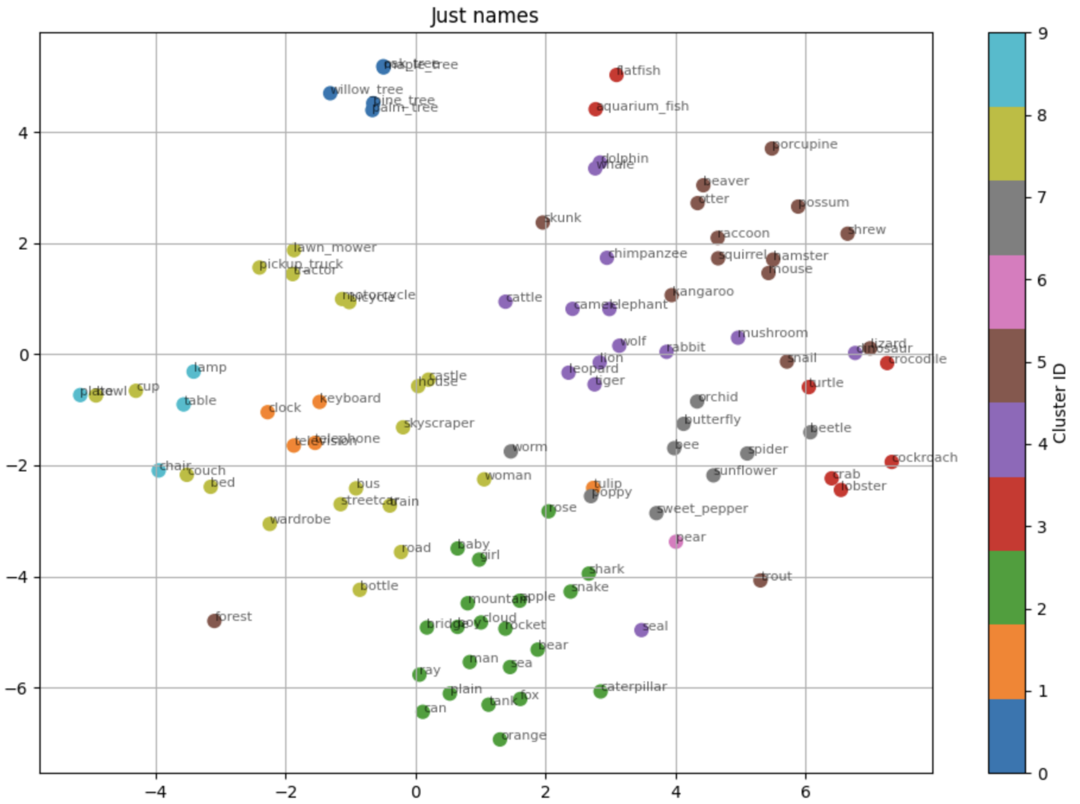


FIGURE A.1: Graphical representation of the text embeddings when using just the class names. The classes are split into clusters using K-means.

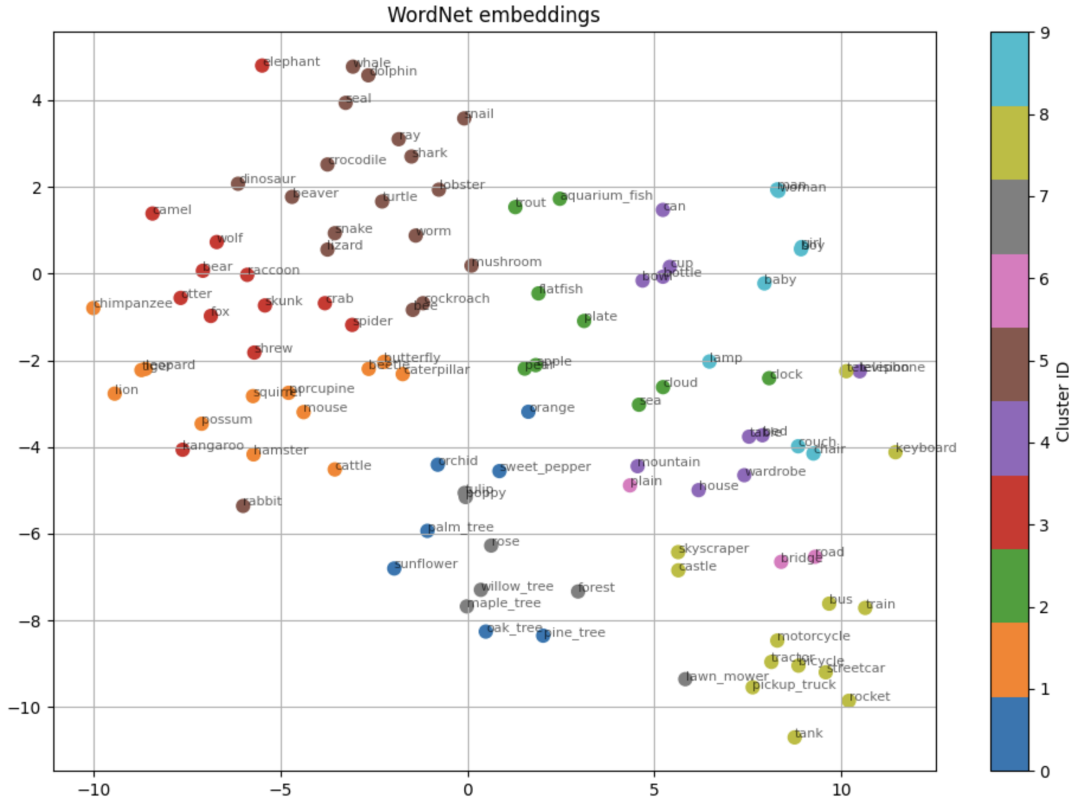


FIGURE A.2: Graphical representation of the text embeddings when using word embeddings. The classes are split into clusters using K-means.

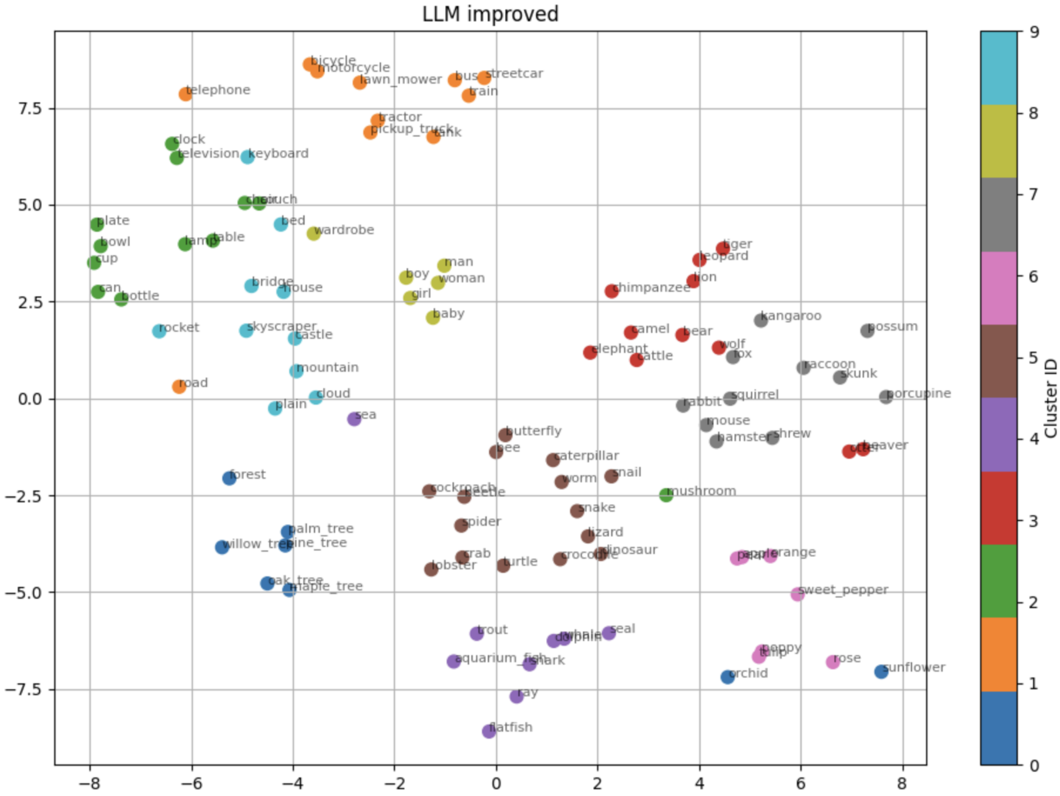


FIGURE A.3: Graphical representation of the text embeddings when using high-quality semantics from an LLM. The classes are split into clusters using K-means.

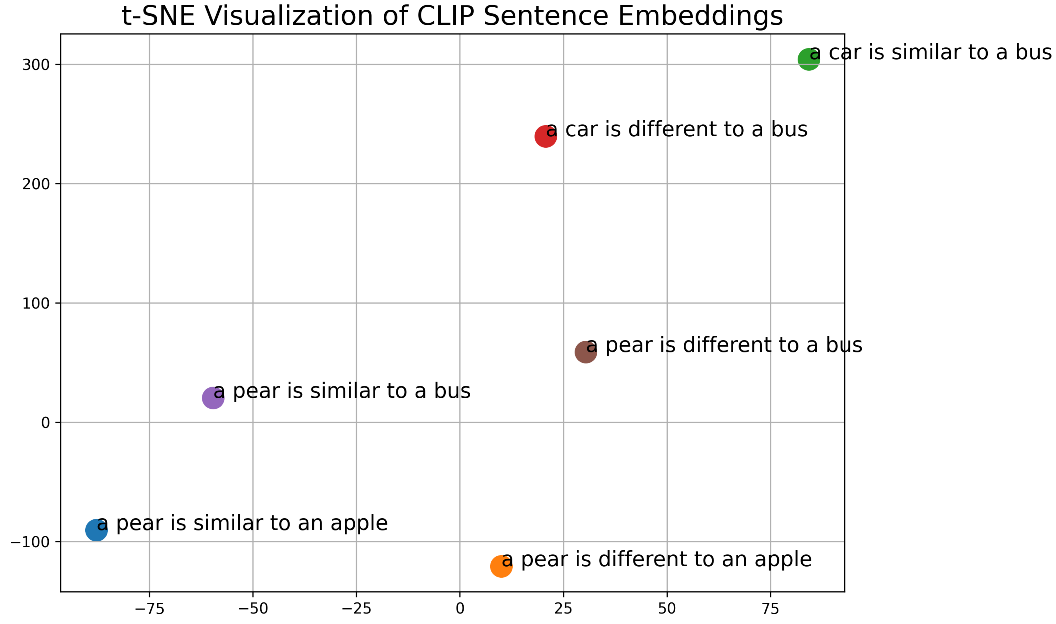


FIGURE A.4: Comparison of different embeddings comparing different classes. It is more important the classes mentioned than the similarity or difference.

Appendix B

Prompt templates

Prompt 0: Definition refinement

Prompt

```
"{definition}" is the definition of the {class}.  Please
rewrite and expand this definition to make it more detailed and
consistent with scientific fact.  Briefness is required, using
only one paragraph.
```

Output

```
"baby":  "A baby, also known as an infant, refers to a very young
child from birth to approximately 1 year of age, characterized
by significant physical and cognitive development.  At this
stage, the infant has not yet begun to walk or talk, relying on
caregivers for mobility and communication.  Typically, infants
start to discover their environment, develop basic motor skills,
and begin to babble and make sounds to initiate communication,
although they may not yet utter recognizable words.  During this
period, they are completely dependent on their caregivers for
nutrition, comfort, and safety, and undergo rapid growth and
development, achieving major milestones such as lifting their
head, rolling over, sitting up, and eventually standing with
support."
```

Commentary

The output relies too heavily on the given wordnet description, expanding on details that are irrelevant to the intended task. Concepts such as "nutrition" and "growth" do not introduce meaningful visual information.

Prompt 1: Guide detail

Prompt

```
Write a short, visually detailed description of the class
'{class}' that will help a vision model correctly recognize
and distinguish it from other classes.  Focus only on unique and
visible traits such as color, shape, texture, size, or key parts.
Do not mention other classes.  Limit to a single paragraph.
```

Output

```
"baby":  "The class 'baby' typically consists of a small, rounded
human figure, usually between 0-2 years old.  They have a
relatively large head compared to their body, with a soft and
smooth texture to their skin.  Their hair is often fine, thin,
and lightly colored, and their facial features are characterized
by large eyes, a small nose, and a mouth with a subtle, gentle
curve.  Their attire often includes onesies or diapers with
distinctive patterns and colors, and they may be accompanied by
pacifiers, toys, or other childcare-related accessories.  The
overall size of a baby is compact, typically under 40 inches in
length, and their movements are often fluid and uncoordinated,
with a characteristic lack of muscular definition."
```

Commentary

In this case, the LLM describes fine-grained details that, although theoretically visual, cannot be perceived in images at this resolution. Moreover, these details tend to be subjective.

Prompt 2: Improved knowledge

Prompt

```
The following is a definition of the class '{class}':
"{definition}".  Using this as context (only if useful), write a
short visual description to help an AI model recognize '{class}'
based on appearance alone.  Do not include other classes.  One
paragraph only.
```

Output

```
"baby":  "A baby typically has a small, rounded body with a large
head in proportion to their body.  Their facial features are soft
and delicate, with a short, button nose and full cheeks.  Their
eyes are usually large and expressive, and their mouth is small
with a subtle curve.  They often have a sparse amount of fine
hair on their head, and their skin tone can vary.  Their limbs
are short and not yet fully developed, with tiny hands and feet,
and they are often unable to control their movements, resulting
in a somewhat floppy posture."
```

Commentary

No comments. The description is the kind of description we are looking for. I think the success from this prompts comes from explicitly saying "only if useful" on the prompt, letting it forget about the wordnet description if deemed necessary, and focusing on the role of being an AI trying to identify images. This prompt actually resulted in the one that obtained the best results when paired with DeepSeek.

Prompt 3: Confused classes

Prompt

```
Define visually the class '{class}', focusing on visual and
physical traits.  Also, focus on the traits that make the
{class} visually distinct from similar categories like {confused
classes}.  Do not mention other classes.  Briefness is required,
using only one paragraph.
```

Output

```
"baby":  ":  The class 'baby' is visually characterized by a
small, rounded body, typically with a chubby face, soft features,
and a smooth, often hairless head.  Their eyes are usually
large in proportion to their face, with a curious and innocent
expression.  They have a distinct physical vulnerability, with
short limbs, a plump torso, and often a reliant posture.  A
diaper or onesie often completes their signature look.  Compared
to similar categories, a baby visually differs from a boy or
girl by their smaller size, softer features, and developmental
immaturity; from an apple by their obvious biological form; from
a man by their tiny stature and facial features; and from a house
by their movable, living form."
```

Commentary

The output mentions too many other classes, causing the model to be confused.

Prompt 4: All classes

Prompt

```
Please write a detailed visual definition of the class {class}.
Make it more visually detailed and consistent with scientific
fact for an AI model to differentiate it from this class to other
classes like {other classes}. Briefness is required, using only
one paragraph.
```

Output

```
"baby": "A baby is a small, typically human primate
characterized by a proportionally large head, large eyes, and
a relatively underdeveloped body compared to adults. Its skin
is smooth and soft, often displaying a range of colors determined
by genetics, and may be covered in fine, downy hair. Limbs are
short and stubby, with tiny hands and feet. Facial features
include a small nose, full cheeks, and a mouth often pursed
or open, suggesting frequent expressions of crying, smiling,
or sucking. The baby's behavior is mainly characterized by
reflexes and a lack of fine motor control, relying heavily on
caregivers for basic survival needs such as feeding, hygiene, and
protection. Its eyes are typically focused forward, reflecting
a binocular vision required for seeing and interpreting facial
expressions from close range."
```

Commentary

It did not pay any attention to the prompt demands. It also commits the same mistakes as Prompt 0.

Prompt 5: Image based

Prompt

```
These 5 images are examples from the class '{class}'. Based on
these images, give a detailed visual description that summarizes
the typical appearance of this class. Focus on the content of
the images, not the format. Briefness is required, using only
one paragraph.
```

Output

```
"baby": "The images show babies of diverse ethnicities, mostly
appearing to be between 6 and 12 months old, with varying hair
colors and textures. They are predominantly shown lying down
or sitting up, often with light clothing or blankets. Facial
expressions range from neutral to slightly smiling. All images
are slightly blurry."
```

Commentary

I included here a particularly poor example to illustrate my point. With this style of prompting, the description becomes highly dependent on the attached images, resulting in either vague statements or overly specific details tied to those particular examples. Moreover, it comments on the form or style of the image rather than its actual content.

Multi Prompt: Iterative refinement

Prompt

```
1. Definition of the class '{class}': {definition}. Expand this
with scientific and visual information about typical appearance
and attributes. Also, contrast this class with the following
visually similar/confused classes: {confused classes}.
2. Here is a preliminary description of the class '{class}':
{previous output}. Now, refine this based on the following 5
example images from the class. Focus only on what is actually
visible and typical in these images. Update the description
accordingly to help an AI model recognize the class visually.
3. Here is a visual description of the class '{cls}': {previous
output} Revise this to remove any mentions to other classes
like {confused classes}. The output should be a long paragraf
that only describe the target class '{class}' and not rely on
comparisons to others.
```

Output

```
"baby": "The class 'baby' refers to a human infant that can
be visually characterized by a distinctive set of features. A
baby typically has a round face with a soft, smooth complexion,
often with a pink or fair skin tone. The face is often the
most prominent feature, with a focus on the eyes and mouth.
The eyes are usually large in proportion to the face, and the
mouth is small, often appearing slightly open. The hair on the
head is typically sparse or absent, and the overall appearance
of the head is rounded and smooth. The body is often soft and
curved, with a subtle definition of the limbs, which may not be
clearly visible. The baby is often depicted wearing clothing
or wrapping, such as diapers or swaddling, which adds to their
distinctive visual appearance. The images of babies often have
a blurred or soft focus, suggesting a close-up shot or low image
quality, and the background may not be clearly visible. Overall,
the visual cues that define a baby include a round face, large
eyes, small mouth, sparse or no hair, and a soft, curved body,
often with a subtle presence of clothing or wrapping. The
combination of these features, along with the characteristic
soft focus and prominent facial features, helps to create a
distinctive visual identity for the class 'baby'."
```

Commentary

This prompt appeared promising but ultimately did not deliver as expected. In this case, the output is overly long and focuses too much on the image's form rather than its actual content. As in previous examples, it seems the model did not fully understand the intended task. I believe this prompt has significant potential; however, since not all of our selected LLMs had access to image input, and those that did faced token limitations, we were unable to explore this approach to its fullest potential.

Bibliography

Alibaba (2024). *Qwen2.5: A Party of Foundation Models*. URL: https://qwenlm.github.io/blog/qwen2.5/.

Andrychowicz, Marcin et al. (2016). “Learning to Learn by Gradient Descent by Gradient Descent”. In: *Advances in Neural Information Processing Systems (NeurIPS) 29*.

Antoniou, Antreas, Harrison Edwards, and Amos Storkey (2019). “How to Train Your MAML”. In: *International Conference on Learning Representations (ICLR)*. DOI: 10.48550/arXiv.1810.09502. URL: https://openreview.net/forum?id=r1lC0yR5K7.

Behl, Harsh, Atılım Güneş Baydin, and Philip Torr (2019). “Alpha MAML: Adaptive Model-Agnostic Meta-Learning”. In: *6th ICML Workshop on Automated Machine Learning, Thirty-Sixth International Conference on Machine Learning (ICML)*. URL: https://arxiv.org/abs/1902.05038.

Bertinetto, Luca et al. (2019). “Meta-learning with differentiable closed-form solvers”. In: *International Conference on Learning Representations (ICLR)*.

Breiman, Leo (1996). “Bagging Predictors”. In: *Machine Learning* 24.2, pp. 123–140. DOI: 10.1007/BF00058655. URL: https://doi.org/10.1007/BF00058655.

— (2001). “Random Forests”. In: *Machine Learning* 45.1, pp. 5–32. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

Chen, Hao et al. (2023a). “Sparse Spatial Transformers for Few-Shot Learning”. In: *Science China Information Sciences* 66.11, p. 210102.

Chen, Tianqi and Carlos Guestrin (2016). “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

Chen, Ting et al. (2020). “A Simple Framework for Contrastive Learning of Visual Representations”. In: *arXiv preprint arXiv:2002.05709*.

Chen, Wentao et al. (2023b). “Semantic Prompt for Few-Shot Image Recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23581–23591.

Chen, Yinbo et al. (2021). “Meta-baseline: Exploring simple meta-learning for few-shot learning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9062–9071.

Chen, Zhihong et al. (2023c). “On the Difference of BERT-style and CLIP-style Text Encoders”. In: *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13710–13721. DOI: 10.18653/v1/2023.findings-acl.866. URL: https://aclanthology.org/2023.findings-acl.866/.

Cheng, Hao et al. (2023). “Frequency guidance matters in few-shot learning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11814–11824.

Cherti, Mehdi et al. (2022). *Reproducible Scaling Laws for Contrastive Language-Image Learning*. URL: https://arxiv.org/abs/2212.07143.

DeepSeek-AI (2025). *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. arXiv: 2501.12948 [cs.CL]. URL: https://arxiv.org/abs/2501.12948.

Deng, Jia et al. (2009). "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 248–255.

Devlin, Jacob et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, pp. 4171–4186. URL: https://aclanthology.org/N19-1423.

Diéguez, Joel (2023). "Exploring Transformers for Localizing Moments of Actions". Directed by Petia Radeva and Estefanía Talavera. Bachelor's thesis. Universitat de Barcelona, Facultat de Matemàtiques. URL: https://hdl.handle.net/2445/198502.

Dong, Bin et al. (2022a). "Self-promoted Supervision for Few-Shot Transformer". In: *European Conference on Computer Vision (ECCV)*, pp. 329–347.

Dong, Bowen et al. (2022b). "Self-Promoted Supervision for Few-Shot Transformer". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, pp. 329–347.

Duan, Yan, John Schulman, Xi Chen, et al. (2016). "RL^2: Fast Reinforcement Learning via Slow Reinforcement Learning". In: *arXiv preprint arXiv:1611.02779*. DOI: 10.48550/arXiv.1611.02779. URL: https://arxiv.org/abs/1611.02779.

Finn, Chelsea, Pieter Abbeel, and Sergey Levine (2017). "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *International Conference on Machine Learning (ICML)*. PMLR, pp. 1126–1135. DOI: 10.5555/3294996.3295055. URL: https://proceedings.mlr.press/v70/finn17a.html.

Freund, Yoav and Robert E. Schapire (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1, pp. 119–139. DOI: 10.1006/jcss.1997.1504. URL: https://doi.org/10.1006/jcss.1997.1504.

Gao, Hongyang and Siyuan Sun (2023). "Meta-AdaM: A meta-learned adaptive optimizer with momentum for few-shot learning". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Google, DeepMind (2024a). *Gemini*. URL: https://deepmind.google/technologies/gemini/.

— (2024b). *Gemma*. URL: https://deepmind.google/models/gemma/.

Hao, Fusheng et al. (2023). "Class-Aware Patch Embedding Adaptation for Few-Shot Image Classification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18905–18915.

He, Kaiming et al. (2016). "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

He, Kaiming et al. (2022). "Masked Autoencoders Are Scalable Vision Learners". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009.

Hiller, Markus et al. (2022). "Rethinking Generalization in Few-Shot Classification". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 35, pp. 3582–3595.

Hou, Ruibing et al. (2019). "Cross attention network for few-shot classification". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Hu, Shuxiao et al. (2022a). "Pushing the Limits of Simple Pipelines for Few-Shot Learning: External Data and Fine-Tuning Make a Difference". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9068–9077.

Hu, Zhuotao, Yujun Sun, and Yi Yang (2022b). "Suppressing the Heterogeneity: A Strong Feature Extractor for Few-Shot Segmentation". In: *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*. Available at https://openreview.net/forum?id=0quX8ZZX0jY.

Kim, Jaekyeom, Hyoungseok Kim, and Gunhee Kim (2020). "Model-Agnostic Boundary-Adversarial Sampling for Test-Time Generalization in Few-Shot Learning". In: *European Conference on Computer Vision (ECCV)*. URL: http://vision.snu.ac.kr/projects/mabas.

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations (ICLR)*. arXiv: 1412.6980 [cs.LG].

Kirkpatrick, James et al. (2017). "Overcoming catastrophic forgetting in neural networks". In: *Proceedings of the National Academy of Sciences* 114.13, pp. 3521–3526. DOI: 10.1073/pnas.1611835114. URL: http://www.pnas.org/content/114/13/3521.abstract.

Koch, Gregory, Richard Zemel, and Ruslan Salakhutdinov (2015). "Siamese Neural Networks for One-Shot Image Recognition". In: *ICML Deep Learning Workshop*. Lille, France.

Krizhevsky, Alex (2009). *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto.

Lee, Kwonjoon et al. (2019). "Meta-learning with differentiable convex optimization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10657–10665.

Lee, Yoonho and Seungjin Choi (2018). "Gradient-Based Meta-Learning with Learned Layerwise Metric and Subspace". In: *International Conference on Machine Learning (ICML)*. PMLR, pp. 2927–2936. DOI: 10.5555/3327144.3327246. URL: http://proceedings.mlr.press/v80/lee18d.html.

Li, Hongyang et al. (2019). "Finding task-relevant features for few-shot learning by category traversal". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–10.

Li, Ke and Jitendra Malik (2017a). "Learning to Optimize". In: *International Conference on Learning Representations (ICLR)*.

Li, Zhan, Feng Zhou, Fei Chen, et al. (2017b). "Meta-SGD: Learning to Learn Quickly for Few-Shot Learning". In: *arXiv preprint arXiv:1707.09835*. DOI: 10.48550/arXiv.1707.09835. URL: https://arxiv.org/abs/1707.09835.

Li, Zhuoling, Yong Wang, and Kaitong Li (2024). "FewVS: A Vision-Semantics Integration Framework for Few-Shot Image Classification". In: *Proceedings of the 32nd ACM International Conference on Multimedia*. MM '24. Melbourne VIC, Australia: Association for Computing Machinery, 1341–1350. ISBN: 9798400706868. DOI: 10.1145/3664647.3681427. URL: https://doi.org/10.1145/3664647.3681427.

Liu, Ze et al. (2021). "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022.

Maas, Andrew L., Awni Y. Hannun, and Andrew Y. Ng (2013). "Rectifier Nonlinearities Improve Neural Network Acoustic Models". In: *Proc. ICML*. Workshop on Deep Learning for Audio, Speech and Language Processing.

Majumder, Orchid et al. (2021). *Supervised Momentum Contrastive Learning for Few-Shot Classification*. arXiv: 2101.11058 [cs.CV]. URL: https://arxiv.org/abs/2101.11058.

Meta (2024). *Introducing Meta Llama 3: The most capable openly available LLM to date*. URL: https://ai.meta.com/blog/meta-llama-3/.

— (2025). *The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation*. URL: https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

Mishra, Nikhil, Mostafa Rohaninejad, Xi Chen, et al. (2018). "A Simple Neural Attentive Meta-Learner". In: *International Conference on Learning Representations (ICLR)*. URL: https://openreview.net/forum?id=rJY0-Kcll.

MistralAI (2025). *Mistral Small 3.1*. URL: https://mistral.ai/news/mistral-small-3-1.

Munkhdalai, Tsendsuren and Hong Yu (2017). "Meta Networks". In: *International Conference on Machine Learning (ICML)*. PMLR, pp. 2554–2563. DOI: 10.5555/3305381.3305548. URL: http://proceedings.mlr.press/v70/munkhdalai17a.html.

Nichol, Alex, Joshua Achiam, and John Schulman (2018). "On First-Order Meta-Learning Algorithms". In: *arXiv preprint arXiv:1803.02999*. DOI: 10.48550/arXiv.1803.02999. URL: https://arxiv.org/abs/1803.02999.

OpenAI (2023). *GPT-4 Technical Report*. URL: https://openai.com/research/gpt-4.

Oreshkin, Boris, Pau Rodríguez López, and Alexandre Lacoste (2018). "TADAM: Task Dependent Adaptive Metric for Improved Few-Shot Learning". In: *Advances in Neural Information Processing Systems*. Vol. 31.

Peng, Zhimao et al. (2019). "Few-shot image recognition with knowledge transfer". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 441–449.

Radford, Alec et al. (2021). "Learning Transferable Visual Models From Natural Language Supervision". In: *International Conference on Machine Learning (ICML)*.

Ravi, Sachin and Hugo Larochelle (2016). "Optimization as a Model for Few-Shot Learning". In: *International Conference on Learning Representations (ICLR)*.

Ruder, Sebastian (2017). *An overview of gradient descent optimization algorithms*. arXiv: 1609.04747 [cs.LG]. URL: https://arxiv.org/abs/1609.04747.

Rusu, Andrei A., Daniel Rao, Jakub Sygnowski, et al. (2019). "Meta-Learning with Latent Embedding Optimization". In: *International Conference on Learning Representations (ICLR)*. DOI: 10.48550/arXiv.1807.05960. URL: https://openreview.net/forum?id=rJY0-Kcll.

Rusu, Andrei A. et al. (2016). "Progressive Neural Networks". In: *arXiv preprint* abs/1606.04671. DOI: 10.48550/arXiv.1606.04671. URL: https://arxiv.org/abs/1606.04671.

Santoro, Adam, Sergey Bartunov, Matthew Botvinick, et al. (2016). "Meta-Learning with Memory-Augmented Neural Networks". In: *International Conference on Machine Learning (ICML)*. PMLR, pp. 1842–1850. DOI: 10.5555/3045390.3045487. URL: http://proceedings.mlr.press/v48/santoro16.html.

Snell, Jake, Kevin Swersky, and Richard S. Zemel (2017). "Prototypical Networks for Few-Shot Learning". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30.

Song, Yisheng et al. (2022). *A Comprehensive Survey of Few-shot Learning: Evolution, Applications, Challenges, and Opportunities*. arXiv: 2205.06743 [cs.LG]. URL: https://arxiv.org/abs/2205.06743.

Sung, Flood, Yongxin Yang, Li Zhang, et al. (2018). “Learning to Compare: Relation Network for Few-Shot Learning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1199–1208.

Tian, Yonglong et al. (2020). *Rethinking Few-Shot Image Classification: a Good Embedding Is All You Need?* arXiv: 2003.11539 [cs.CV]. URL: https://arxiv.org/abs/2003.11539.

Tsoumplekas, Georgios et al. (2025). *A Complete Survey on Contemporary Methods, Emerging Paradigms and Hybrid Approaches for Few-Shot Learning*. arXiv: 2402.03017 [cs.LG]. URL: https://arxiv.org/abs/2402.03017.

Tyukin, Ivan et al. (July 2021). *Demystification of Few-shot and One-shot Learning*. DOI: 10.1109/IJCNN52387.2021.9534395.

Vaswani, Ashish et al. (2017). “Attention is All You Need”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. DOI: 10.5555/3295222.3295349. URL: https://arxiv.org/abs/1706.03762.

Vinyals, Oriol et al. (2016). “Matching Networks for One Shot Learning”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 29, pp. 3630–3638.

Wang, Xiaolong et al. (2022). “Instruction Tuning”. In: *arXiv preprint* abs/2211.03378. DOI: 10.48550/arXiv.2211.03378. URL: https://arxiv.org/abs/2211.03378.

Wang, Xin et al. (2019). “TAFE-Net: Task-Aware Feature Embeddings for Low Shot Learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1831–1840.

Wang, Yaqing et al. (2020). *Generalizing from a Few Examples: A Survey on Few-Shot Learning*. arXiv: 1904.05046 [cs.LG]. URL: https://arxiv.org/abs/1904.05046.

Wei, Jason et al. (2022). “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”. In: *arXiv preprint* abs/2201.11903. DOI: 10.48550/arXiv.2201.11903. URL: https://arxiv.org/abs/2201.11903.

Wolpert, David H. (1992). “Stacked Generalization”. In: *Neural Networks* 5.2, pp. 241–259. DOI: 10.1016/S0893-6080(05)80023-1. URL: https://doi.org/10.1016/S0893-6080(05)80023-1.

Xing, Chen et al. (2019). “Adaptive cross-modal few-shot learning”. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Xu, Jingyi and Hieu Le (2022). “Generating representative samples for few-shot classification”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9003–9013.

Yan, An et al. (2022). “CLIP also Understands Text: Prompting CLIP for Phrase Understanding”. In: *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. URL: https://arxiv.org/abs/2210.05836.

Yan, Kun et al. (2021). “Aligning Visual Prototypes with BERT Embeddings for Few-Shot Learning”. In: *Proceedings of the International Conference on Multimedia Retrieval (ICMR)*, pp. 367–375.

Ye, Han-Jia et al. (2020). “Few-Shot Learning via Embedding Adaptation with Set-to-Set Functions”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8808–8817.

Zhang, Hai et al. (2024). “Simple Semantic-Aided Few-Shot Learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28588–28597.

Zhang, Hao et al. (2022). *DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection*. arXiv: 2203.03605 [cs.CV].

Zhou, Jing et al. (2022). “Automatic Prompt Generation”. In: *arXiv preprint* abs/2203.09196. DOI: 10.48550/arXiv.2203.09196. URL: https://arxiv.org/abs/2203.09196.