

ALERT: ADAPTIVE LEARNING WITH EMBEDDINGS AND REINFORCEMENT FOR TRANSPARENT FEW-SHOT LEARNING

Anonymous authors

Paper under double-blind review

ABSTRACT

In this paper, we introduce ALERT, a novel approach to few-shot learning that significantly enhances both the interpretability and accuracy of large vision-language models (LVLMs) in classification tasks with limited data. By utilizing the strengths of LVLMs and integrating a meta-task instruction framework, ALERT effectively transforms the traditional black-box nature of few-shot models into a transparent process. It allows for traceable and understandable reasoning. ALERT employs learnable category embeddings to emphasize unique features of each category, improving classification accuracy, and introduces a contrastive reward function within a Group Relative Policy Optimization (GRPO) training framework to enhance reasoning capabilities and training stability. Our extensive experiments across various datasets demonstrate that ALERT consistently outperforms existing few-shot learning methods, achieving state-of-the-art results. Notably, in the 16-shot setting on ImageNet, ALERT achieved an impressive accuracy of 78.74%, significantly improving on previous methods.

1 INTRODUCTION

In recent years, the field of computer vision has witnessed tremendous advancements, largely driven by the development of sophisticated models such as convolutional neural networks and transformers. These technologies have demonstrated extraordinary success across a spectrum of vision tasks, especially when applied to large-scale datasets. However, the challenge of limited data availability in certain scenarios has shifted focus towards few-shot learning, a burgeoning area of research. Few-shot learning involves training models to discern complex patterns and make accurate predictions even when presented with a minimal amount of labeled data.

A recent breakthrough in this domain is CLIP (Radford et al., 2021a), a paradigm-shifting approach that leverages large-scale language-image pairs for pre-training, demonstrating robust zero-shot transfer capabilities for open-vocabulary visual recognition tasks. Building on CLIP’s foundation, subsequent models (Zhou et al., 2022; Gao et al., 2021; Zhang et al., 2021) have extended this zero-shot framework to few-shot classification, achieving remarkable performance across various datasets. These advancements highlight the potential of leveraging large-scale pre-training to empower networks with superior representation abilities, even when few-shot training data is scarce.

Despite these significant strides, current models often function as black boxes, producing a score for each class without offering insight into the decision-making process. This opacity is a critical limitation in applications that require transparency and understanding of model decisions. While these models achieve high accuracy, the lack of interpretability remains a persistent challenge.

In this paper, we introduce ALERT, a novel approach to few-shot learning that enhances both the reasoning capabilities and accuracy of models. Unlike traditional models, ALERT is built on the promising developments in Large Vision Language Models (LVLMs), which are designed to integrate the complementary information present in both vision and language inputs. By employing a meta-task instruction framework, ALERT transforms the traditional black-box nature of few-shot learning models into a more transparent process where reasoning can be traced and understood. As illustrated in Figure 1, our method provides more detailed analysis and identifies subtle similarities and differences, leading to more accurate classifications.

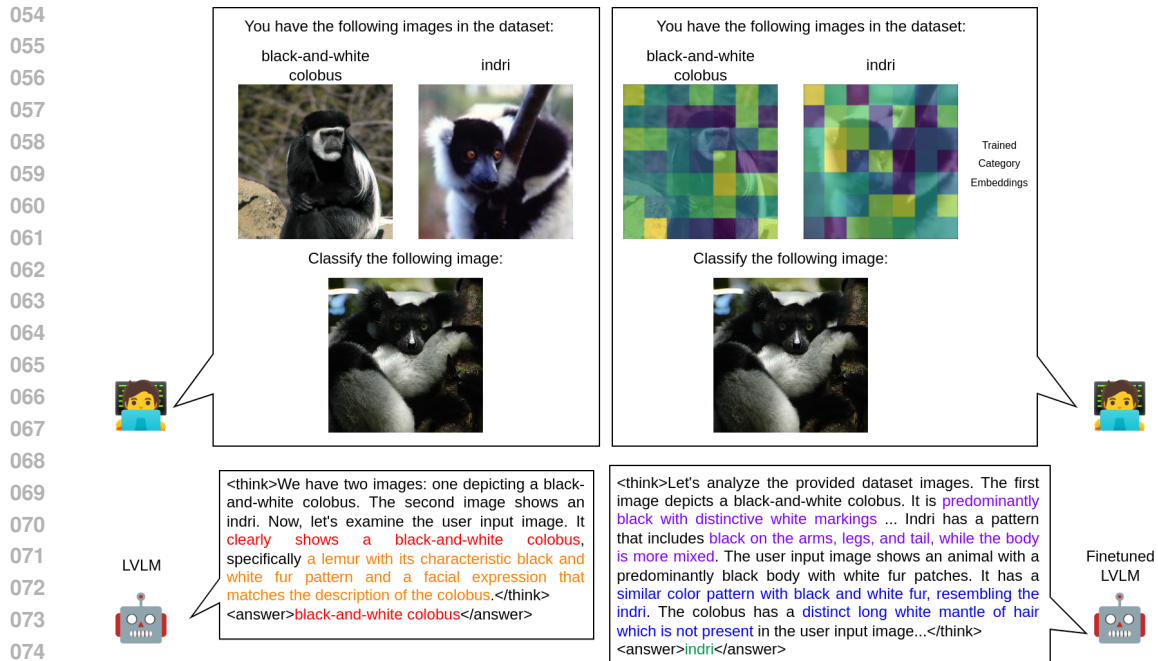


Figure 1: This example showcases our finetuned result compared with the pretrained LVM. The pretrained LVM (left) has premature conclusions. In contrast, the finetuned LVM (right) provides a more detailed analysis, identifying subtle similarities and differences such as the longer white mantle of hair and mixed body pattern, leading to the correct classification.

Our contributions can be summarized as follows:

1. We propose learnable category embeddings that emphasize specific features unique to each category, improving the model’s classification accuracy.
2. We introduce a contrastive reward function within a Group Relative Policy Optimization (GRPO) framework, enhancing the model’s reasoning capabilities and stability.
3. We demonstrate the effectiveness of an explainable few-shot learning pipeline utilizing candidate selection and a meta-task instruction framework, which guides the model in generating clear and logical reasoning for classification tasks.

2 RELATED WORKS

2.1 FEW-SHOT LEARNING

Few-shot learning is a classification paradigm that focuses on the ability of models to generalize from a limited number of training examples, often as few as one or two per class. Unlike traditional machine learning approaches that require large amounts of labeled data, few-shot learning aims to mimic the human ability to learn new concepts quickly with minimal data. Typically, this is achieved by leveraging prior knowledge gained from related tasks or datasets, enabling the model to adapt rapidly to new tasks with minimal additional training.

Prominent strategies include meta-learning, which trains models to adapt quickly to new tasks (Chen et al., 2021; Finn et al., 2017; Jamal & Qi, 2019; Li et al., 2021), and metric learning, which constructs a feature space where classification is based on proximity (Snell et al., 2017; Sung et al., 2018; Vinyals et al., 2016). However, these approaches often lack interpretability, making it difficult to understand their decision-making process.

108 More recently, vision-language models like CLIP (Radford et al., 2021a), pre-trained on web-scale
109 data, have shown significant promise. Adaptation techniques such as prompt tuning in CoOp (Zhou
110 et al., 2022) and lightweight adapters in CLIP-Adapter (Gao et al., 2021) further enhance their per-
111 formance on downstream few-shot tasks. Despite these advancements, a common criticism remains:
112 the lack of transparency in how these models arrive at their decisions, which is crucial for applica-
113 tions requiring explainability.

114 Our approach addresses these weaknesses by integrating a meta-task instruction framework within
115 a Large Visual Language Model (LVLM) to provide interpretability alongside high accuracy.
116

117 2.2 LVLM INSTRUCTION TUNING

119 LVLMs represent a significant advancement in vision-language integration. Inspired by success-
120 ful instruction tuning in LLMs, recent work has focused on enhancing LVLMs with instruction-
121 following data to improve performance across diverse tasks. For instance, LLaVA-1.5 (Liu et al.,
122 2023) improved on Visual Question Answering, InstructBLIP (Dai et al., 2023) enhanced zero-shot
123 capabilities, and Gemma 3 (Team et al., 2025) refined dialogue interactions through such tuning.

124 Our approach stands out by incorporating learnable category embeddings during the LVLM tun-
125 ing process, unlike typical methods that rely only on text and images. This innovation enhances
126 classification accuracy and interpretability, allowing the model to make more nuanced comparisons
127 in few-shot learning scenarios. By combining instruction tuning with category embeddings, our
128 method fully utilizes the LVLM’s potential for precise and interpretable few-shot classification.
129

130 2.3 GROUP RELATIVE POLICY OPTIMIZATION (GRPO)

132 GRPO (Shao et al., 2024) is a reinforcement learning technique that has been instrumental in ad-
133 vancing the reasoning capabilities of LLMs. This method was notably utilized in the DeepSeek-R1
134 model (DeepSeek-AI, 2025), which showcased the potential of pure RL strategies in fostering com-
135 plex reasoning patterns within LLMs. A pivotal discovery from DeepSeek-R1 is the ability of GRPO
136 to induce the emergence of chain-of-thought reasoning and "aha moments," where models actively
137 re-evaluate and correct their reasoning processes.

138 The GRPO approach involves generating a group of outputs for each training sample from an ex-
139 isting policy. It calculates advantages using pre-defined reward functions for each output, guiding
140 the policy update towards more refined reasoning. This optimization strategy enables models to de-
141 velop reasoning capabilities by reinforcing outputs that demonstrate superior logical coherence and
142 accuracy.

143 In our work, we adapt the GRPO optimization strategy from DeepSeek-R1 to enhance few-shot
144 learning tasks. This adaptation involves proposing a contrastive reasoning reward function to im-
145 prove training stability and accuracy. By incorporating this reward function, we aim to bolster the
146 model’s ability to discern subtle differences and similarities within few-shot learning scenarios, thus
147 enhancing its interpretability and performance.
148

149 3 METHODOLOGY

150 3.1 CATEGORY CANDIDATE SELECTION

153 In the initial stage of our few-shot learning pipeline, we can employ any established few-shot learn-
154 ing method for selecting potential category candidates. For each given image, we utilize a few-shot
155 learning model as a lightweight draft model to generate a logit for every category within the dataset.
156 Each logit quantifies the likelihood that the image belongs to a particular category. We then select the
157 top- k categories with the highest logits as candidate categories. These selected categories are incor-
158 porated into the instruction prompt for the Large Visual Language Model (LVLM) during the second
159 stage of our pipeline. In this stage, the LVLM is tasked with choosing the most likely category for
160 the image during the inference process. This methodology effectively reduces task complexity and
161 enhances the LVLM’s self-consistency (Liu et al., 2025) by narrowing down the range of possible
categories, thereby facilitating more accurate classification.

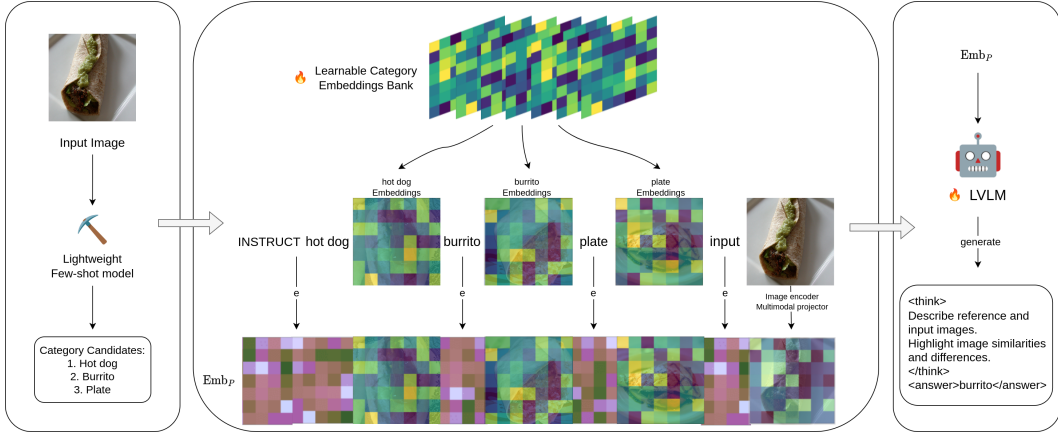


Figure 2: This diagram illustrates our ALERT few-shot learning pipeline. It begins with an input image, which is processed by a lightweight few-shot model to select category candidates. The embeddings of these categories are then combined with the input image embedding and text embeddings to form a meta-task instruction prompt. This prompt is fed into the LVLM, which generates reasoning and a classification answer.

3.2 TRAINABLE CATEGORY EMBEDDING

Each category within the dataset has its own trainable 2D embedding matrix. Mathematically, the embedding for category c_i can be denoted as $\text{Emb}_{c_i} \in \mathbb{R}^{m \times d}$ where m is the number of tokens per image embedding and d is the embedding dimension. For each category, we first extract image embeddings from the few-shot training images using the LVLM’s image encoder. These image embeddings are then averaged to form an initial representative embedding for the category. This averaged embedding is subsequently passed through the LVLM’s multimodal projector to create the initialized category embedding. These initialized category embeddings are stored in a category embeddings bank, and we let the model access and update them during training.

3.3 META-TASK INSTRUCTION

We have developed a meta-task instruction framework to prompt the LVLM into generating responses. This framework requires reference categories from the dataset to be included within the instruction prompt. When classifying a given image, the LVLM selects the most likely reference category from a set of reference categories, denoted as $\{r_{c_1}, r_{c_2}, \dots, r_{c_k}\}$, where k represents the number of reference classes. The meta-task instruction prompt P is structured as follows:

$$T_{\text{instruct}} \oplus r_{c_1} \oplus \text{Emb}_{c_1} \oplus \dots \oplus r_{c_k} \oplus \text{Emb}_{c_k} \oplus \text{Emb}_{\text{input}}$$

where \oplus signifies the concatenation operation. T_{instruct} is the text instruction designed to guide the LVLM in classifying the given image. The terms Emb_{c_i} represent the category embeddings of the reference classes, which are retrieved from the category embeddings bank. $\text{Emb}_{\text{input}} \in \mathbb{R}^{m \times d}$ represents the image embedding of the input image, which is extracted using the LVLM’s image encoder and multimodal projector. The complete meta-task instruction prompt is detailed in Appendix A.

During training, we randomize the number of reference categories provided in the instruction prompt. One of the dataset category labels is the ground-truth image label c_{gt} , while other labels are sampled from all available dataset categories based on the similarity between the ground-truth category images and those of other categories.

We begin by calculating the cosine similarity between the L2-normalized average image embeddings from the few-shot training images of each category, resulting in a logit that quantifies the similarity

216 between the ground-truth category c_{gt} and each other category in the dataset:
217

$$218 \quad l_{c_{gt}, c_b, k} = \left(\frac{\sum_{i=1}^k f_{c_{gt}}^i / k}{\left\| \sum_{i=1}^k f_{c_{gt}}^i / k \right\|} \right)^T \left(\frac{\sum_{i=1}^k f_{c_b}^i / k}{\left\| \sum_{i=1}^k f_{c_b}^i / k \right\|} \right)$$

219 Here, $l_{c_{gt}, c_b, k}$ represents the similarity logit for the k -shot learning scenario between category c_{gt}
220 and another category c_b . The term $f_{c_{gt}}^i$ denotes the image embedding extracted from the i -th image
221 in category c_{gt} , and $f_{c_b}^i$ denotes the image embedding from the i -th image in category c_b .

222 Reference categories are sampled based on the computed weights:
223

$$224 \quad w_{c_{gt}, c_b, k} = \frac{\exp(l_{c_{gt}, c_b, k} / \tau)}{\sum_{j=1}^C \exp(l_{c_{gt}, c_j, k} / \tau)}$$

225 In this equation, τ is a temperature parameter (less than 1) used to adjust the likelihood of selecting
226 categories that are more similar to the ground-truth label. A lower τ value increases the difficulty of
227 the training samples by making the selection of similar categories more pronounced.

228 During inference, the reference categories are the top- k classes from the category candidate selec-
229 tion.
230

231 3.4 TEXT GENERATION FROM EMBEDDING

232 To process the instruction text part of our meta-task instruction prompt P , we begin by tokenizing
233 the instruction text. These text tokens are then converted into their corresponding embeddings,
234 denoted as Emb_P . Throughout this process, we retain all category embeddings and the input image
235 embedding in their original form. The embedding sequence is structured as follows:
236

$$237 \quad \text{Emb}_P = [e(T_{\text{instruct}, 1}), \dots, e(T_{\text{instruct}, I}),$$

$$238 \quad e(r_{c_1, 1}), \dots, e(r_{c_1, C_1}), \text{Emb}_{c_1}, \dots,$$

$$239 \quad e(r_{c_k, 1}), \dots, e(r_{c_k, C_k}), \text{Emb}_{c_k}, \text{Emb}_{\text{input}}]$$

240 where $\{T_{\text{instruct}, 1}, \dots, T_{\text{instruct}, I}\}$ represent tokens derived from T_{instruct} , $\{r_{c_1, 1}, \dots, r_{c_1, C_1}\}$ are tokens
241 from r_{c_1} , $\{r_{c_k, 1}, \dots, r_{c_k, C_k}\}$ are tokens from r_{c_k} , and $e(\cdot)$ is the token embedding function for
242 mapping tokens to their respective embeddings.
243

244 The next token logits can be obtained by

$$245 \quad H = \text{Transformer}(\text{Emb}_P)$$

$$246 \quad z = W \cdot h / T$$

247 where $H \in \mathbb{R}^{M \times d}$ is the matrix of the last hidden states for all tokens up to position M , $h = H[M, :]$
248 is the last hidden state of position M , W is the parameter of the language model head and T is the
249 sampling temperature.
250

251 Finally, the next token is sampled from the probability distribution derived from the tempera-
252 ture-scaled logits z . This sampled token is then converted into its corresponding embedding and ap-
253 pended to the sequence, allowing the LVLm to continue generating the response based on the initial
254 instruction prompt and input. This process repeats for each subsequent token until an end condition,
255 such as a specific token indicating the end of the response, is met.
256

257 3.5 GRPO REWARD FUNCTIONS

258 Reward functions used in GRPO reinforcement learning score the text completions generated by a
259 model based on pre-defined criteria. We employ four reward functions in GRPO training:
260

- 261 1. Strict Format Reward Function: This function verifies whether the completion adheres to
262 the required format by enclosing reasoning within `<think>` and `</think>`, and the
263 predicted label within `<answer>` and `</answer>`. This approach is inspired by the
264 format reward function used in DeepSeek-R1 (DeepSeek-AI, 2025).
265

- 270 2. Soft Format Reward Function: This function evaluates whether the completion at least
271 partially aligns with the reasoning format.
- 272 3. Accuracy Reward Function: This function checks if the predicted label enclosed within
273 `<answer>` and `</answer>` precisely matches the ground-truth label.
- 274 4. Contrastive Reasoning Reward Function: We use an LLM as a judge to give a score to the
275 generated reasoning text. The score is based on whether the generated reasoning is able to
276 identify similarities and differences between the provided image and dataset images, high-
277 light relevant details, and ensure the class decision is justified by the identified similarities
278 and differences. The full prompt used for this LLM judge is provided in Appendix B. This
279 reward function can improve the RL training stability.

282 4 EXPERIMENTS

284 4.1 EXPERIMENTAL SETTINGS

286 4.1.1 DATASETS

288 Following previous works (Zhou et al., 2022; Zhang et al., 2021; 2023), we perform few-shot experi-
289 ments on 11 publicly available datasets: ImageNet (Deng et al., 2009), StanfordCars (Krause et al.,
290 2013), UCF101 (Soomro et al., 2012), Caltech101 (Fei-Fei et al., 2004), Flowers102 (Nilsback &
291 Zisserman, 2008), SUN397 (Xiao et al., 2010), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al.,
292 2019), FGVC Aircraft (Maji et al., 2013), OxfordPets (Parkhi et al., 2012), and Food101 (Bossard
293 et al., 2014). Specifically, we train ALERT under the few-shot setups with 1, 2, 4, 8, 16 shots and
294 test on full test splits in all datasets.

296 4.1.2 TRAINING DETAILS

297 We use the instruction-tuned Gemma 3 (Team et al., 2025) 4B model as our pretrained LVLM. To
298 enhance training efficiency and reduce costs, we load the model in 8-bit precision and freeze both
299 the LLM and visual encoder, employing LoRA (Hu et al., 2021) to fine-tune the model’s adapter for
300 4 epochs.

302 The learning process utilizes a cosine learning rate scheduler with a base learning rate of 5×10^{-6}
303 and a warm-up ratio of 0.01. Optimization is performed using the AdamW optimizer (Loshchilov
304 & Hutter, 2019), with a weight decay of 0.1, β_1 set to 0.9, and β_2 set to 0.99, ensuring stable
305 convergence.

306 During training, we randomize the number of reference categories, ranging from 2 to 8. For refer-
307 ence category selection, we set the temperature parameter $\tau = 0.1$ to increase the difficulty of the
308 training samples.

309 We use vLLM library (Kwon et al., 2023) to speed up the text generation process in GRPO training.
310 We configure the number of generations per training sample to be 6 and set the maximum completion
311 length for each generation to 512 tokens. The sampling temperature T is set to 0.9. We evaluate
312 text completions using the following reward function scoring scheme to guide the model’s learning
313 process:

- 314 1. Strict Format Reward Function: Assigns a score of 0.5 if the completion strictly adheres to
315 the specified format, with reasoning enclosed in `<think>` tags and the predicted label in
316 `<answer>` tags; otherwise, the score is 0.
- 317 2. Soft Format Reward Function: Provides a score of 0.5 for partial compliance with the
318 required format, scoring 0 if these criteria are not met.
- 319 3. Accuracy Reward Function: Awards a score of 2 if the predicted label matches the ground-
320 truth label exactly, otherwise 0.
- 321 4. Contrastive Reasoning Reward Function: Uses the Gemma 3 4B LLM to evaluate reasoning
322 quality, scoring from 1 to 10, normalized to a 0 to 1 scale.

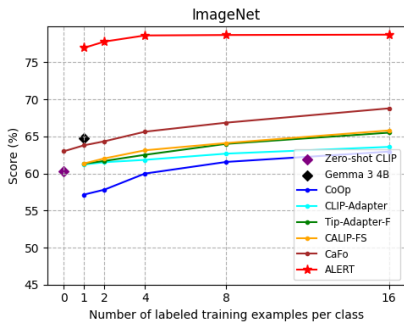


Figure 3: Performance (%) Comparison on ImageNet.

Shot	0	1	2	4	8	16
Zero-shot CLIP	60.33	-	-	-	-	-
Gemma 3 4B	-	64.71	-	-	-	-
Linear-probe CLIP	-	22.17	31.90	41.20	49.52	56.13
CoOp	-	57.15	57.81	59.99	61.56	62.95
CLIP-Adapter	-	61.20	61.52	61.84	62.68	63.59
VT-CLIP	-	60.53	61.29	62.02	62.81	63.92
Tip-Adapter-F	-	61.32	61.69	62.52	64.00	65.51
CALIP-FS	-	61.35	62.03	63.13	64.11	65.81
CaFo	62.99	63.80	64.34	65.64	66.86	68.79
ALERT	-	76.95	77.80	78.63	78.69	78.74

Table 1: Quantative Performance (%) Comparison on ImageNet.

4.1.3 BASELINES

In line with previous studies (Zhang et al., 2023), our baseline comparisons include zero-shot CLIP (Radford et al., 2021a) and several recent few-shot learning methods, such as Linear-probe CLIP (Radford et al., 2021b), CoOp (Zhou et al., 2022), CLIP-Adapter (Gao et al., 2021), VT-CLIP (Qiu et al., 2023), Tip-Adapter-F (Zhang et al., 2021), CALIP-FS (Guo et al., 2022), and CaFo (Zhang et al., 2023). These models are evaluated based on their accuracies on each dataset.

In addition to these established baselines, we introduce the pretrained Gemma 3 4B model as a new baseline. For the Gemma 3 4B model, instead of using trained category embeddings, we randomly select one image per category from the training dataset and utilize its image features extracted by Gemma’s image encoder. This setup is considered a 1-shot setting. The model then generates reasoning and an answer for each test image, and we use the answer as the prediction to compare against the ground-truth class label.

4.2 PERFORMANCE

4.2.1 ON IMAGENET

We compare ALERT performance with our baselines on the most representative ImageNet in Figure 3 and Table 1. Throughout our experiments, unless otherwise specified, we chose to use the top-2 candidates selected from CaFo during the category candidate selection stage. In the 16-shot configuration, ALERT attains an accuracy of 78.74%, notably surpassing the zero-shot CLIP baseline at 60.33% and the one-shot Gemma 3 baseline at 64.71%. Compared to other few-shot learning models, ALERT consistently demonstrates superior accuracy, with enhancements ranging from 14.46% to 20.92% over CaFo across various shot settings. Impressively, ALERT with just 1 shot outperforms all other methods in every setting.

378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431

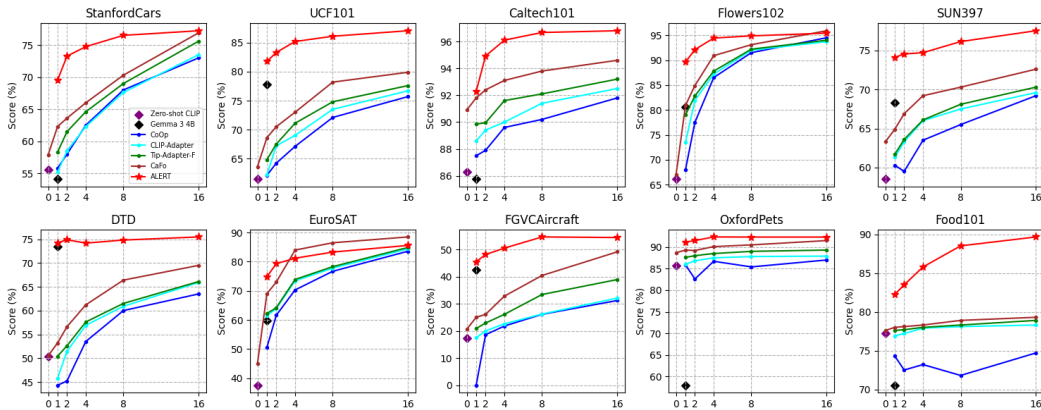


Figure 4: Performance (%) Comparison on 10 Datasets.

Datasets	Source		Target	
	ImageNet	-V2	-Sketch	
Zero-shot CLIP	60.33	53.27	35.44	
Gemma 3 4B	64.71	64.87	57.69	
CoOp	62.95	54.58	31.04	
CLIP-Adapter	63.59	55.69	35.68	
CALIP-FS	65.81	55.98	35.37	
Tip-Adapter-F	65.51	57.11	36.00	
CaFo	68.79	57.99	39.43	
ALERT	78.74	76.70	69.12	

Table 2: **Distribution Shift (%) Comparison.** We train the models on Source dataset and test on Target datasets.

4.2.2 ON OTHER DATASETS

To further evaluate ALERT’s robustness across different scenarios, we tested it on an additional 10 datasets, as detailed in Figure 4. These datasets, which cover a wide range of semantic categories such as real-world scenes, detailed textures, and satellite-captured landscapes, allowed us to assess how well ALERT adapts to diverse data characteristics. The results demonstrate that ALERT consistently outperforms other few-shot learning models.

For instance, on the OxfordPets dataset, ALERT achieved an accuracy of 92.3%, surpassing the performance of CaFo, which reached 91.5%. In the Caltech101 dataset, which includes diverse object categories, ALERT excelled with an impressive accuracy of 96.8%, outperforming CaFo’s 94.6%. The model’s superiority also extends to more specialized domains. On the DTD texture dataset, ALERT achieved an accuracy of 75.5% (vs. CaFo’s 69.5%), and on Food101, it scored 89.7%, significantly outperforming CaFo’s best result of 79.3%.

While ALERT demonstrates competitive performance across most datasets, it underperforms compared to CaFo on EuroSAT. Specifically, ALERT’s accuracy tops at 85.6% in the 16-shot scenario, whereas CaFo achieves 88.5%. This discrepancy indicates that while ALERT excels in general few-shot learning tasks, there is room for improvement in handling satellite image classifications.

4.2.3 DISTRIBUTION SHIFT

To further assess ALERT’s robustness to distribution shifts, we conducted experiments where the model was trained on a Source dataset and then evaluated on Target datasets. In our study, as pre-

sented in Table 2, we used ImageNet as the Source dataset, while ImageNet-V2 and ImageNetSketch served as the Target datasets.

The results demonstrate ALERT’s superior performance over all baselines under distribution shifts. It achieved 76.70% accuracy on ImageNet-V2 and 69.12% on ImageNet-Sketch, showcasing its strong adaptability. These improvements indicate that ALERT effectively inherits valuable prior knowledge from the pre-trained LVLm, enabling it to generalize well to new and unseen data distributions and maintain high performance.

4.3 ABLATION STUDIES

4.3.1 CATEGORY EMBEDDINGS BANK

To evaluate the impact of trainable versus frozen category embeddings, we conducted an ablation study focusing on the Category Embeddings Bank. In this study, we compared the performance of ALERT using trainable category embeddings against a version with frozen embeddings. The results, as shown in Table 3, indicate that the trainable embeddings significantly enhance the model’s performance, achieving an accuracy of 78.74% compared to 69.81% with the frozen embeddings. This improvement underscores the ability of trained embeddings to effectively capture category-specific features and distinguish them from features of other categories, thereby leading to more accurate classification outcomes.

Embedding Type	Accuracy (%)
Frozen Category Embeddings	69.81
Trainable Category Embeddings	78.74

Table 3: Performance (%) Comparison of ALERT with Frozen vs. Trainable Category Embeddings.

4.3.2 CONTRASTIVE REASONING REWARD FUNCTION

We conducted an ablation study to assess the necessity of the contrastive reasoning reward function in our model’s training process. Without this reward function, we observed significant training instabilities after approximately 0.5 epochs, leading to accuracy ranging only between 65.8% and 76.4% on ImageNet. By including the contrastive reasoning reward function, we enhanced training stability and achieved a much higher accuracy of 78.74%. Throughout our experiments, we experienced no stability issues when the contrastive reasoning reward function was used, even with extended training periods. This highlights the critical role of the contrastive reasoning reward function in stabilizing the training process and improving the model’s overall performance.

Configuration	Accuracy (%)
Without Contrastive Reward	65.8 - 76.4
With Contrastive Reward	78.74

Table 4: Performance (%) Comparison of ALERT with and without the Contrastive Reasoning Reward Function on ImageNet.

5 CONCLUSION

We introduced ALERT, a novel few-shot learning approach that enhances both the interpretability and accuracy of LVLms. By combining a meta-task instruction framework with learnable category embeddings and a contrastive reasoning reward function within a GRPO framework, ALERT makes the classification process transparent and traceable. Extensive experiments across 11 datasets show that ALERT consistently outperforms existing methods, achieving state-of-the-art accuracy. Ablation studies confirmed that our proposed trainable embeddings and contrastive reward function are critical for enhancing performance and training stability.

REFERENCES

- 486
487
488 Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative com-
489 ponents with random forests. In *European conference on computer vision*, pp. 446–461, 09 2014.
490 ISBN 978-3-319-10598-7. doi: 10.1007/978-3-319-10599-4_29.
- 491
492 Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-baseline: Explor-
493 ing simple meta-learning for few-shot learning. In *2021 IEEE/CVF International Conference on*
494 *Computer Vision (ICCV)*, pp. 9042–9051, 2021. doi: 10.1109/ICCV48922.2021.00893.
- 495
496 Mircea Cimpoi, Subhansu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. De-
497 scribing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*,
pp. 3606–3613, 2014. doi: 10.1109/CVPR.2014.461.
- 498
499 Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang,
500 Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language
501 models with instruction tuning, 2023.
- 502
503 DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning,
504 2025. URL <https://arxiv.org/abs/2501.12948>.
- 505
506 Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hier-
507 archical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*,
pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.
- 508
509 Li Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training ex-
510 amples: An incremental bayesian approach tested on 101 object categories. In *2004 Con-*
511 *ference on Computer Vision and Pattern Recognition Workshop*, pp. 178–178, 2004. doi:
10.1109/CVPR.2004.383.
- 512
513 Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation
514 of deep networks. In *Proceedings of the 34th International Conference on Machine Learning -*
Volume 70, ICML’17, pp. 1126–1135. JMLR.org, 2017.
- 515
516 Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li,
517 and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint*
arXiv:2110.04544, 2021.
- 518
519 Ziyu Guo, Renrui Zhang, Longtian Qiu, Xianzheng Ma, Xupeng Miao, Xuming He, and Bin Cui.
520 Calip: Zero-shot enhancement of clip with parameter-free attention, 2022. URL [https://](https://arxiv.org/abs/2209.14169)
521 arxiv.org/abs/2209.14169.
- 522
523 Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset
524 and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected*
Topics in Applied Earth Observations and Remote Sensing, 12(7):2217–2226, 2019. doi: 10.
525 1109/JSTARS.2019.2918242.
- 526
527 Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,
528 and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL [https:](https://arxiv.org/abs/2106.09685)
529 [//arxiv.org/abs/2106.09685](https://arxiv.org/abs/2106.09685).
- 530
531 Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In
532 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11711–
11719, 2019. doi: 10.1109/CVPR.2019.01199.
- 533
534 Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained
535 categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pp. 554–
536 561, 2013. doi: 10.1109/ICCVW.2013.77.
- 537
538 Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E.
539 Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model
serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating*
Systems Principles, 2023.

- 540 Bohao Li, Boyu Yang, Chang Liu, Feng Liu, Rongrong Ji, and Qixiang Ye. Beyond max-margin:
541 Class margin equilibrium for few-shot object detection, 2021. URL [https://arxiv.org/
542 abs/2103.04612](https://arxiv.org/abs/2103.04612).
- 543 Fan Liu, Wenwen Cai, Jian Huo, Chuanyi Zhang, DeLong Chen, and Jun Zhou. Making large vision
544 language models to be good few-shot learners. *Proceedings of the AAAI Conference on Artificial
545 Intelligence*, 39:5415–5423, 04 2025. doi: 10.1609/aaai.v39i5.32576.
- 546 Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.
- 547 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL [https://
548 //arxiv.org/abs/1711.05101](https://arxiv.org/abs/1711.05101).
- 549 Subhansu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained
550 visual classification of aircraft, 2013. URL <https://arxiv.org/abs/1306.5151>.
- 551 Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number
552 of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*,
553 pp. 722–729, 2008. doi: 10.1109/ICVGIP.2008.47.
- 554 Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In
555 *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012. doi:
556 10.1109/CVPR.2012.6248092.
- 557 Longtian Qiu, Renrui Zhang, Ziyu Guo, Ziyao Zeng, Zilu Guo, Yafeng Li, and Guangnan Zhang.
558 Vt-clip: Enhancing vision-language models with visual-guided texts, 2023. URL [https://
559 arxiv.org/abs/2112.02399](https://arxiv.org/abs/2112.02399).
- 560 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agar-
561 wal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya
562 Sutskever. Learning transferable visual models from natural language supervision, 2021a. URL
563 <https://arxiv.org/abs/2103.00020>.
- 564 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agar-
565 wal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya
566 Sutskever. Learning transferable visual models from natural language supervision, 2021b. URL
567 <https://arxiv.org/abs/2103.00020>.
- 568 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
569 Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathe-
570 matical reasoning in open language models, 2024. URL [https://arxiv.org/abs/2402.
571 03300](https://arxiv.org/abs/2402.03300).
- 572 Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning.
573 In *Proceedings of the 31st International Conference on Neural Information Processing Sys-
574 tems, NIPS’17*, pp. 4080–4090, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN
575 9781510860964.
- 576 Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions
577 classes from videos in the wild, 2012. URL <https://arxiv.org/abs/1212.0402>.
- 578 Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales.
579 Learning to compare: Relation network for few-shot learning. In *2018 IEEE/CVF Conference
580 on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018. doi: 10.1109/CVPR.2018.
581 00131.
- 582 Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej,
583 Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivièrè, Louis Rouillard, Thomas
584 Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Cas-
585 bon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xi-
586 aohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Cole-
587 man, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry,
588 Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi,

594 Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe
 595 Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa
 596 Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András
 597 György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia
 598 Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petri
 599 Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel
 600 Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Pappas, Divyashree Shivaku
 601 mar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eu
 602 gene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna
 603 Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian
 604 Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wi
 605 eting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh,
 606 Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine,
 607 Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael
 608 Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Ni
 609 lay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Ruben
 610 stein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya
 611 Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu,
 612 Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti
 613 Sheth, Siim Pöder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi
 614 Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry,
 615 Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein
 616 Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat
 617 Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas
 618 Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Bar
 619 ral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam
 620 Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena
 621 Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier
 622 Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot.
 623 Gemma 3 technical report, 2025. URL <https://arxiv.org/abs/2503.19786>.

622 Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Match
 623 ing networks for one shot learning. In *Proceedings of the 30th International Conference on Neural*
 624 *Information Processing Systems, NIPS' 16*, pp. 3637–3645, Red Hook, NY, USA, 2016. Curran
 625 Associates Inc. ISBN 9781510838819.

627 Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database:
 628 Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference*
 629 *on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010. doi: 10.1109/CVPR.2010.
 630 5539970.

631 Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hong
 632 sheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv*
 633 *preprint arXiv:2111.03930*, 2021.

634 Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Hongsheng Li, Yu Qiao, and
 635 Peng Gao. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot
 636 learners. *arXiv preprint arXiv:2303.02151*, 2023.

638 Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for
 639 vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*
 640 *(CVPR)*, 2022.

643 A META-TASK INSTRUCTION PROMPT

644
 645 In our meta-task instruction prompt shown in Listing 1, the placeholders <Reference
 646 category label>, <Reference category embedding>, and <Image embedding>
 647 are replaced with specific values during the process. Specifically, <Reference category 1
 label> is replaced with the category label r_{c_1} , <Reference category 1 embedding> is

Listing 1 Prompt for our meta-task instruction

You are an image classifier. You are given a dataset containing images with their corresponding labels. When a user provides an input image, your task is to classify it into one of the existing classes. Follow these steps:

1. Carefully examine and describe the dataset images, noting key details and characteristics of each class.
2. Analyze and describe the user input image in detail.
3. Compare the user input image with the dataset images:
 - Identify and highlight similarities between the input image and the images of each class.
 - Identify and highlight differences between the input image and the images not belonging to the chosen class.
4. Use your observations to determine the most appropriate class for the user input image, ensuring your reasoning is clear and logical. Then provide your classification decision.

The reasoning process and answer are enclosed within `<think></think>` and `<answer></answer>` tags, respectively, i.e., `<think>reasoning process here</think><answer>answer here</answer>`.

Dataset images:

Class name: `<Reference category 1 label> <Reference category 1 embedding>`

Class name: `<Reference category 2 label> <Reference category 2 embedding>`

...

User input image: `<Image embedding>`

Listing 2 Prompt for scoring the reasoning

Evaluate the student’s reasoning for classifying an image based on the following criteria and provide a score from 1 to 10:

- Identification of similarities between the given image and the dataset images of the chosen class.
- Identification of differences between the given image and the dataset images not belonging to the chosen class.
- Identification of relevant details in both the given image and the dataset images.
- The student claims the class only after clearly highlighting the similarities and differences.

The student’s reasoning:

`<reasoning>`

Output only the score (1-10) without formattings and explanations.

replaced with the corresponding category embedding Emb_{c_1} , and `<Image embedding>` is replaced with the image embedding $\text{Emb}_{\text{input}}$. These substitutions ensure that the LVLM receives all necessary information to accurately classify the input image by comparing it with the reference categories and their embeddings, which are included in the instruction prompt.

B CONTRASTIVE REASONING JUDGE PROMPT

The contrastive reasoning judge prompt shown in Listing 2 is designed to evaluate the quality of the LVLM’s reasoning when classifying an image. It instructs the evaluator to score the reasoning on a scale from 1 to 10 based on several criteria: the identification of similarities between the given image and dataset images of the chosen class, the identification of differences with images not belonging to the chosen class, and the identification of relevant details in both the given image and the dataset images. The prompt emphasizes that the LVLM should only claim the class after clearly highlighting these similarities and differences. In this prompt, the placeholder `<reasoning>` is replaced with the reasoning text generated during GRPO training, allowing for an objective assessment of the classification process.

C VISUALIZATION OF CLASSIFICATION AND REASONING

Figure 5 provides additional visualizations that illustrate the detailed reasoning process of our model during the classification task on examples from the ImageNet dataset. This figure showcases two examples, demonstrating the model’s capability to handle varying numbers of reference categories effectively.

D ADDITIONAL ABLATION STUDIES

D.1 CATEGORY SELECTION COUNT

To understand the impact of the number of category candidates on ALERT’s performance, we conducted an ablation study varying the number of top candidates selected during the classification process. The selection of top category candidates is crucial for narrowing down the possible classes, which can significantly affect the model’s ability to make accurate predictions. We evaluated ALERT using different numbers of top category candidates: top-2, top-3, top-4, and top-8.

Number of Top Candidates	Accuracy (%)
Top-2 Candidates	78.74
Top-3 Candidates	76.63
Top-4 Candidates	75.23
Top-8 Candidates	72.85

Table 5: Performance (%) of ALERT with Different Numbers of Top Category Candidates on ImageNet.

The results are summarized in Table 5. The study indicates that selecting the top-2 candidates results in the highest accuracy of 78.74%. This suggests that focusing on the most relevant categories during inference allows the model to maintain high precision without the distraction of too many options.

D.2 CATEGORY EMBEDDINGS INITIALIZATION

Initialization Method	Accuracy (%)
Random	76.20
Averaging Method	77.63
Proposed Method	78.74

Table 6: Performance (%) Comparison of Different Initialization Methods for Category Embeddings.

In our experiments, we explored different initialization strategies for category embeddings to assess their impact on the performance of our model. We tested two additional methods alongside our primary approach: random initialization and an averaging method that combines embeddings after the image encoder layer and the multimodal projector layer. The random initialization involves setting the category embeddings to random values, while the averaging method computes the mean of embeddings obtained from both the image encoder and the multimodal projector layers.

Our results in Table 6 show that the choice of initialization method can affect the model’s performance, albeit slightly. The random initialization achieved an accuracy of 76.20%, while the averaging method resulted in a 77.63% accuracy. These findings indicate that while initialization strategies can influence the outcomes, their overall effect is relatively minor compared to other factors in the model design and training process.

756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

Reference categories:



Input image:



<think>Let's analyze the dataset images first. We have an Arctic fox, an Alaskan tundra wolf, a Samoyed, and a polar bear. The Arctic fox is characterized by its thick, dense white fur and relatively small size – it looks like a fluffy, compact animal. The Alaskan tundra wolf is larger with a more rugged appearance and darker coloring around the eyes and muzzle. The Samoyed is distinctly characterized by its extremely fluffy white coat, giving it a cloud-like appearance. Finally, the polar bear is the largest of the group, with a thick layer of white fur and a distinctive orange nose. Now, let's examine the user input image. It shows a white animal with thick fur, lying down. This animal has rounded ears, a relatively small nose, and a gentle expression. Comparing the user input image to the dataset images, it most closely resembles the Arctic fox. Both have thick white fur and a compact, rounded appearance. The user input image also portrays a lying-down posture, similar to the Arctic fox in the dataset.</think>
<answer>Arctic fox</answer>

Reference categories:



Input image:



<think>Let's analyze the dataset images first. We have several lizard species represented: agama, desert grassland whiptail lizard, green iguana, alligator lizard, frilled-necked lizard, Gila monster, chameleon, Komodo dragon. The images show various colors, patterns, and sizes of lizards. The agama image shows a brightly colored lizard on a rock, while the whiptail lizard is a smaller, slender lizard on bare ground. The green iguana is large and heavily armored, and the alligator lizard is similar in size and shape to the whiptail, but with different coloration. The frilled-necked lizard has a distinctive frill around its neck, the Gila monster is pink and patterned, the chameleon is a small, camouflaged lizard, and the Komodo dragon is a very large, powerful reptile. Now, let's examine the user input image. The image shows two lizards on a rocky surface, similar to the whiptail lizard. They are of similar size and coloration. Considering these similarities, the most likely classification for the user input image is 'desert grassland whiptail lizard'.</think>
<answer>desert grassland whiptail lizard</answer>

Figure 5: Additional visualizations demonstrating candidate selection with 4 categories (top) and 8 categories (bottom). The above examples are from the ImageNet dataset. The images of the reference categories are for illustration purpose only. We use trained category embeddings instead of individual images.