This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Incentivizing Generative Zero-Shot Learning via Outcome-Reward Reinforcement Learning with Visual Cues

Wenjin Hou[1] Xiaoxiao Sun[2] Hehe Fan[1,3*]
[1]CCAI, Zhejiang University [2]Stanford University
[3]State Key Laboratory of CAD&CG, Zhejiang University
houwj17@gmail.com xxsun@stanford.edu hehefan@zju.edu.cn

Abstract

Recent advances in zero-shot learning (ZSL) have demonstrated the potential of generative models. Typically, generative ZSL synthesizes visual features conditioned on semantic prototypes to model the data distribution of unseen classes, followed by training a classifier on the synthesized data. However, the synthesized features often remain task-agnostic, leading to degraded performance. Moreover, inferring a faithful distribution from semantic prototypes alone is insufficient for classes that are semantically similar but visually distinct. To address these and advance ZSL, we propose RLVC, an outcome-reward reinforcement learning RL framework with visual cues for generative ZSL. At its core, RL empowers the generative model to self-evolve, implicitly enhancing its generation capability. In particular, RLVC updates the generative model using an outcome-based reward, encouraging the synthesis of task-relevant features. Furthermore, we introduce class-wise visual cues that (i) align synthesized features with visual prototypes and (ii) stabilize the RL training updates. For the training process, we present a novel cold-start strategy. Comprehensive experiments and analyses on three prevalent ZSL benchmarks demonstrate that RLVC achieves state-of-the-art results with a 4.7% gain.

1. Introduction

Generative models (*e.g.*, variational autoencoders (VAEs)) [39], generative adversarial networks (GANs) [27], and diffusion models (DMs) [30]) have emerged as practical solutions for zero-shot learning (ZSL) [4, 33, 75]. Using predefined semantic prototypes (*e.g.*, expert-annotated attribute vectors [12, 34, 84, 87] or word embeddings of class names [16, 53, 55, 73]) as conditions, these models synthesize high-quality visual features [17, 33, 40] or images [21, 25, 31, 42] for unseen classes. As a result, they offer an unbounded representation space for modeling data distributions in generative ZSL, effectively alleviating the lack of unseen classes.

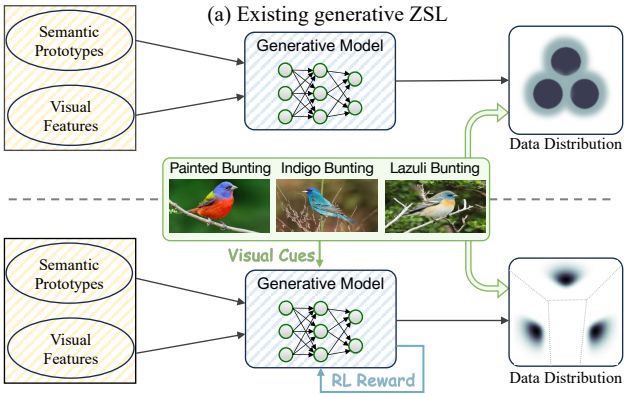


Figure 1. Motivating illustration. (a) Existing generative ZSL methods train with adversarial losses conditioned only on semantic prototypes. This often leads to task-agnostic synthesized features and inter-class overlap. (b) Our RLVC incentivizes the generative model updating via RL reward and visual cues, enabling synthesized features that remain task-relevant and faithfully represent the data distribution.

To improve synthesized samples and advance generative ZSL, existing ZSL solutions mainly pursue four directions: **i**) fine-tuning the visual backbone on the seen classes of the ZSL dataset [56, 76, 87]. **ii**) enforcing visual–semantic consistency via bidirectional mapping [32, 46]. **iii**) learning enhanced visual features for classifier training [6, 10, 17, 40]. **iv**) evolving semantic prototypes with visual features for better alignment [12, 33, 45, 87]. Rather than aligning visual and semantic features from scratch, recent approaches, such as CLIP [61] and SHIP [73], which utilize large-scale vision-language pre-training, also provide suitable class prototypes. Although impressive, the synthesized features obtained via these methods exhibit two limitations: **Firstly**, the generative model is typically optimized independently of the downstream classifier, which restricts its ability to

*Corresponding author.

model task-relevant data distributions. **Secondly**, some methods rely exclusively on the semantic conditions, leading to overlapping inter-class feature distributions and misclassifications. For example, the classes "Indigo Bunting", "Lazuli Bunting" and "Painted Bunting" are semantically similar but visually distinct (see Fig. 1). Hence, we posit that richer supervision is necessary to bolster tolerance to inter-class variance.

In light of the above, we present **RLVC**, an outcome-reward reinforcement learning (RL) framework with class-wise visual cues for generative ZSL. At its core, RL mimics human trial-and-error learning to achieve goals through self-evolution (*i.e.*, learning how to take actions to maximize reward) [52, 62]. Based on these advantages, we consider RL's outcome-based optimization well suited for improving the capability of generative models (the *"Why"*). Moreover, RL can align the generation process more directly with the downstream classification objective, rather than relying on adversarial losses tied only to semantic prototypes, thereby balancing inter-class separation and task relevance. In addition, class-wise visual cues act as reliable supervision during the training process.

Specifically, we treat the generator (denoted as G_θ) as a policy model from the view of RL [1]. For task-relevant generation, we design a classifier to serve as the reward model. We pre-train this reward model to produce an outcome-based score (*i.e.*, the predicted probability of the given class) as the reward signal. This reward drives updates of G_θ, explicitly aligning synthesized features with the downstream classification task. Meanwhile, we mine class-wise visual cues from fine-tuned visual features of seen classes and take them as visual prototypes. Then, we impose a prototype-distillation loss to directly align synthesized features with these visual prototypes. As an additional benefit, visual cues also stabilize RL optimization.

In terms of the training paradigm, we adopt a novel cold-start strategy. We first perform some iterations using the generative loss, then activate RL training once a preset threshold is reached. To alleviate gradient conflict and improve optimization stability, we employ an alternating mechanism at each iteration (*i.e.*, we update G_θ separately with the generative adversarial loss and the RL loss). In addition, we fine-tune the visual features to mitigate cross-dataset bias and enhance generalization. Consequently, RLVC synthesizes features that faithfully represent the data distribution and remain task-relevant (Fig. 1(b)), effectively alleviating the above issues.

To sum up, our core contributions can be concluded as:

- **Novel perspective**. We present RLVC from an RL viewpoint. To our knowledge, this is the first attempt to analyze and apply RL to generative ZSL.
- **Controlled framework**. We introduce *an outcome-reward reinforcement-learning framework with visual cues* to incentivize the generative model. We further present a novel cold-start training recipe to stabilize the optimization process.
- **Empirical validation**. We conduct systematic empirical validation across three critical ZSL benchmark datasets. Our experiments reveal that RLVC significantly *outperforms* current state-of-the-art (SOTA) ZSL methods.

[1]Throughout the paper, "generative model", "generator", and "policy model" are interchangeable terms in our method.

2. Related Works

2.1. Zero-Shot Learning

The goal of ZSL is to recognize unseen classes by knowledge learned from seen ones [43, 58]. Classical solutions are typically categorized into two paradigms according to the ultimate classification space. The earlier one is projection methods (*i.e.*, embedding methods), which directly map visual features to the semantic space with a transformation function supervised by semantic prototypes [1, 5, 13, 15, 18, 19, 22, 34, 35, 50, 54, 55, 70, 79, 81]. The second paradigm can be viewed as a data augmentation strategy (*i.e.*, generative methods). It synthesizes visual features to represent the data distribution of unseen classes. Then, training a classifier in the visual space to perform ZSL [2, 4, 6, 7, 11, 12, 17, 25, 29, 33, 66, 75, 84, 87]. For embedding methods, recent advances mainly learn locally aligned visual–semantic correspondences via attention mechanisms. For instance, TransZero++ [8, 9] and ZS-LViT [14] employ transformer-based cross-modal attention to align visual patches with attributes. PSVMA+ [48, 49] and VSPCN [37] enforce progressive visual–semantic mutual alignment. Despite these advancements, the learned embedding space of this line of work often exhibits bias toward seen classes [86]. Besides, under weak semantic conditions (*e.g.*, class names), achieving fine-grained local alignment remains challenging [16, 77]. Therefore, generative methods have received increasing attention recently.

Within the generative paradigm, early methods focus on bidirectional semantic↔visual consistency with simple decoders and ℓ_1/ℓ_2 losses, such as cycle-CLSWGAN [23], FREE [6], and TFVAEGAN [56]. LisGAN [44] and Lsr-GAN [69] take a classification loss as a part of the discriminator loss. Subsequent work improves training strategies: CE-GZSL [29] and ICCE [40] learn contrastive embeddings; ZLAP [5] adjusts logits; ESZSL [4] proposes sample probing, SC-EGG [32] adopts an embedding-guided generator; and ZeroNAS [84] introduces NAS into ZSL. More recent advances, such as TDCSS [24], DSP [12], VADS [33], GenZSL [16], ViFR [17], and ZeroDiff [87], inject stronger priors and model unseen class–conditional distributions more faithfully [88]. Despite this progress, two structural issues persist: **i**) most approaches optimize

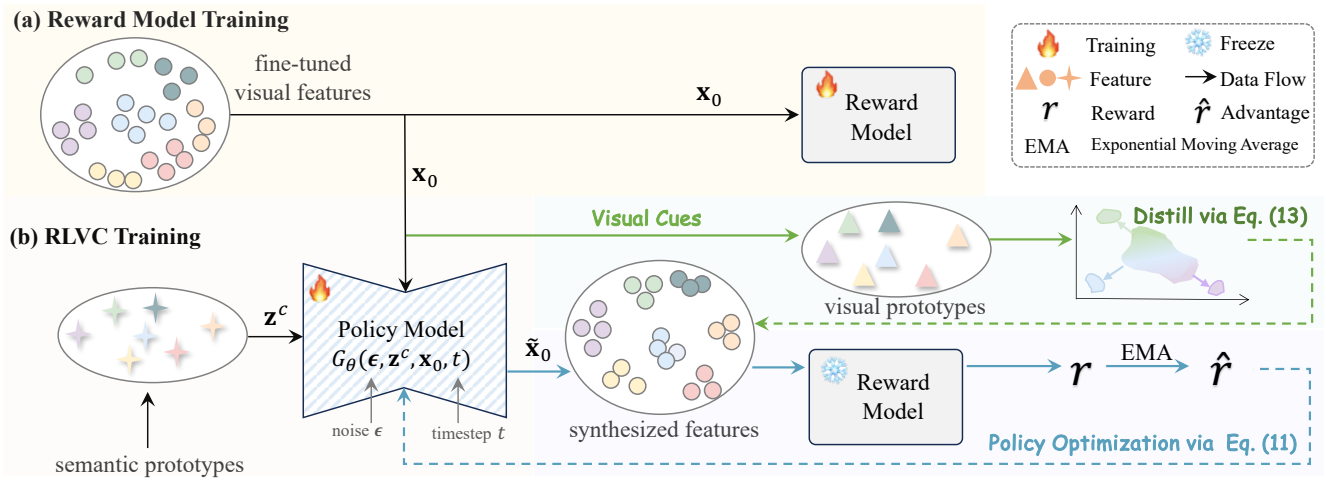


Figure 2. Model architecture and training of RLVC. The top panel shows how we train the reward model with a visual encoder to produce fine-tuned visual features and reward signals. The bottom panel depicts how we update the policy model G_θ (*i.e.*, generator) via outcome-reward reinforcement learning (blue arrows) and visual cues (green arrows), enabling synthesized features that remain task-relevant and faithfully represent the data distribution. $\mathbf{x}_0$ and $\tilde{\mathbf{x}}_0$ denote the real and synthesized features of seen classes, respectively.

the generator independently of the downstream classifier, yielding *task-agnostic* features; and **ii**) semantic-only conditions often induce *inter-class confusion*, especially for fine-grained categories that are semantically similar but visually distinct. Inspired by the success of RL in vision, we propose an RL framework to improve the generator's modeling capability. Meanwhile, we mine class-wise visual cues to distill richer visual information for feature synthesis.

2.2. Reinforcement Learning for Visual Tasks

RL optimizes decision-making by interacting with an environment to maximize cumulative reward. Its applications span gaming, embodied AI, finance, and puzzle [26, 64, 65, 67]. More recently, RL has played an increasingly important role in post-training of large language models (LLMs) to enhance reasoning ability [41]. Frontier models such as DeepSeek-R1 [28], OpenAI o1 [36], and Qwen [71, 85] employ verifiable rewards to optimize task performance, encouraging models to align with human intentions [51, 72].

Motivated by RL's ability to self-evolve in the language domain and its strong generalization, the vision community has begun to explore effective RL frameworks. For example, Visual-RFT [52] and VPRL [83] introduce visual reinforcement fine-tuning (RFT) and extend RFT to visual tasks. Broadly, these efforts either design task-grounded rewards or feedback (*e.g.*, accuracy rewards, format rewards [52]) or develop more efficient optimization strategies (*e.g.*, DPO [80], GRPO [63]). Building on these insights, we make the first attempt to investigate whether RL can advance ZSL. Rather than proposing complex policy optimization algorithms, we aim to provide a simple pipeline based on outcome-reward RL for faithful visual synthesis in ZSL.

3. RLVC

Motivation and Overview. In this section, we present the core design of RLVC. Our objective is to extend the generative model's modeling capacity to construct data representations well-suited for classification. To this end, we treat the generator as a policy and optimize it via RL. A key component in RLVC is a reward model that delivers task-specific reward signals, enabling stronger model optimization. Within a coherent framework, RLVC first trains the reward model together with the visual encoder. During policy training, the fine-tuned visual features are used to mine visual cues, and the reward model produces an outcome reward (see Fig. 2). We detail these designs and the training procedure in §3.2, §3.3 and §3.4.

Problem Statement. Let $\mathcal{Y}^s$ and $\mathcal{Y}^u$ denote the disjoint sets of seen and unseen classes, respectively, with $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset$ and $C^s = |\mathcal{Y}^s|$, $C^u = |\mathcal{Y}^u|$. Training data are available only for seen classes: $\mathcal{D}^{tr} = \{(\mathbf{x}_i^s, y_i^s, \mathbf{z}^{y_i^s})\}_{i=1}^{N_{tr}}$, where $\mathbf{x}_i^s \in \mathcal{X}$ is a visual feature, $y_i^s \in \mathcal{Y}^s$ is its label, and $\mathbf{z}^c \in \mathcal{A}$ denotes the semantic prototype for class c (*e.g.*, an attribute vector or a text embedding). Semantic prototypes $\{\mathbf{z}^c\}_{c \in \mathcal{Y}^s \cup \mathcal{Y}^u}$ are assumed to be available for both seen and unseen classes. In the conventional ZSL (CZSL) setting, the test set contains only unseen classes $\mathcal{D}^{te,u} = \{(\mathbf{x}_j^u, y_j^u, \mathbf{z}^{y_j^u})\}_{j=1}^{N_u}$, and the goal is to learn a classifier $f_{\text{CZSL}} : \mathcal{X} \to \mathcal{Y}^u$. In the generalized ZSL (GZSL) setting, the test set includes both seen and unseen samples, $\mathcal{D}^{te} = \mathcal{D}^{te,s} \cup \mathcal{D}^{te,u}$ with $\mathcal{D}^{te,s} \subseteq \mathcal{X} \times \mathcal{Y}^s$ and $\mathcal{D}^{te,u} \subseteq \mathcal{X} \times \mathcal{Y}^u$, and the goal becomes $f_{\text{GZSL}} : \mathcal{X} \to \mathcal{Y}^s \cup \mathcal{Y}^u$. The objective in both settings is to leverage $\mathcal{D}^{tr}$ and the semantic prototypes $\{\mathbf{z}^c\}$ to minimize the expected classification error on the corresponding test domain.

3.1. Diffusion-based Generative Framework

Motivated by the strong generative capacity of diffusion models, we adopt a diffusion-based adversarial framework [78, 87]. For simplicity, the framework comprises a generator G_θ and two discriminators, D_{x_0} and D_{x_t}. Given a class prototype $\mathbf{z}^c$, Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, and a diffusion state $\mathbf{x}_t$ at timestep t, the generator outputs a visual feature:

$$\tilde{\mathbf{x}}_0 \;=\; G_\theta\big(\boldsymbol{\epsilon},\, \mathbf{z}^c,\, \mathbf{x}_t,\, t\big) \;\in\; \mathbb{R}^d. \tag{1}$$

Conditioned on $\mathbf{z}^c$, the two discriminators operate as follows. D_{x_0} distinguishes real clean features ($\mathbf{x}_0$) from synthesized ones ($\tilde{\mathbf{x}}_0$). D_{x_t} contrasts the real transition ($\mathbf{x}_t$, $\mathbf{x}_{t+1}$) with the synthesized transition ($\tilde{\mathbf{x}}_t$, $\mathbf{x}_{t+1}$), where $\tilde{\mathbf{x}}_t$ is obtained from $\tilde{\mathbf{x}}_0$ and $\mathbf{x}_{t+1}$ via a first-order posterior update. The optimization objectives are as follows:

$$\mathcal{L}_{D_{x_0}} = -\mathbb{E}\big[D_{x_0}(\mathbf{x}_0, \mathbf{z}^c)\big] + \mathbb{E}\big[D_{x_0}(\tilde{\mathbf{x}}_0, \mathbf{z}^c)\big] + \lambda_{\text{gp}}\, \text{GP}_{x_0}, \tag{2}$$

$$\begin{aligned}\mathcal{L}_{D_{x_t}} \;=\; &-\mathbb{E}\big[D_{x_t}(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{z}^c, t)\big] \\ &+ \mathbb{E}\big[D_{x_t}(\tilde{\mathbf{x}}_t, \mathbf{x}_{t+1}, \mathbf{z}^c, t)\big] + \lambda_{\text{gp}}\, \text{GP}_{x_t},\end{aligned} \tag{3}$$

$$\mathcal{L}_D = \mathcal{L}_{D_{x_0}} + \mathcal{L}_{D_{x_t}}, \tag{4}$$

$$\mathcal{L}_G^{\text{adv}} = -\mathbb{E}\big[D_{x_0}(\tilde{\mathbf{x}}_0, \mathbf{z}^c)\big] - \mathbb{E}\big[D_{x_t}(\tilde{\mathbf{x}}_t, \mathbf{x}_{t+1}, \mathbf{z}^c, t)\big], \tag{5}$$

where λ_{gp} is the gradient-penalty weight and GP denotes the gradient penalty. During training, we alternately update the discriminator objective in Eq. (4) and the generator objective in Eq. (5).

3.2. Outcome-Reward Reinforcement Learning

In generative ZSL, a core challenge is to align generative capacity with discriminative representations that downstream classifiers can reliably use. However, purely adversarial objectives often synthesize task-agnostic features. To bridge this gap, we introduce an *outcome-reward reinforcement learning (RL)* to directly incentivize the generator toward task-relevant synthesis. From the RL perspective, we optimize the policy via self-evolving updates to favor features that are more likely to be correctly classified. Concretely, a frozen classifier R serves as the reward model. R is implemented as a linear layer, *i.e.*, $R(\mathbf{x}) = W\mathbf{x} + \mathbf{b}$ with $W \in \mathbb{R}^{C\times d}$, mapping a d-dim visual feature to C class logits. Given a synthesized feature $\tilde{\mathbf{x}}_0$, R outputs logits that are converted to probabilities via the $\mathrm{softmax}$ operation,

$$p(y \mid \tilde{\mathbf{x}}_0) \;=\; \mathrm{softmax}\big(R(\tilde{\mathbf{x}}_0)\big)_y, \tag{6}$$

and we calculate the log-probability of the ground-truth class as the outcome reward r:

$$r \;=\; \log p(y \mid \tilde{\mathbf{x}}_0)\,. \tag{7}$$

Intuitively, higher confidence of R on the correct class for $\tilde{\mathbf{x}}_0$ yields a larger r, which in turn steers the generator's updates accordingly.

Furthermore, to stabilize training and improve performance, we use an exponential moving average (EMA) baseline b over mini-batch rewards to compute the advantage $\hat{r}_i$. This process is formally defined as:

$$b \;\leftarrow\; \alpha\, b \;+\; (1-\alpha)\,\frac{1}{B}\sum_{i=1}^{B} r_i, \tag{8}$$

$$\hat{r}_i \;=\; r_i \;-\; b, \tag{9}$$

where B is the batch size and $\alpha \in [0, 1)$ controls smoothing (in experiments, we set $\alpha = 0.9$). Moreover, we employ a stop-gradient operator $\mathrm{sg}[\cdot]$ for $\hat{r}_i$, so that it is treated as a constant without gradients and define the final advantage as:

$$\widehat{A}_i = \mathrm{sg}[\hat{r}_i]. \tag{10}$$

Finally, the RL objective $\mathcal{L}_{\text{RL}}$ for the policy model is:

$$\mathcal{L}_{\text{RL}} \;=\; -\,\frac{1}{B}\sum_{i=1}^{B} \widehat{A}_i \;\log p(y_i \mid \tilde{\mathbf{x}}_{0,i})\,. \tag{11}$$

Since $\log p(y_i \mid \tilde{\mathbf{x}}_{0,i})$ is differentiable with respect to $\tilde{\mathbf{x}}_{0,i}$, gradients propagate through R and Eq. (6) to $\tilde{\mathbf{x}}_{0,i}$ and then to G_θ (while keeping R's parameters fixed). This outcome-based reward drives the policy to enhance its generation capability with clear guidance, enabling more discriminative, task-relevant synthesis in the feature space.

3.3. Visual Cues with Prototype-Distillation Loss

Although semantic prototypes provide strong conditions, they are inadequate to faithfully represent data distributions for classes that are semantically similar but visually distinct. Additionally, it is common to include a Kullback–Leibler (KL) regularizer in the objective to constrain distributional shift and stabilize training in RL. Building on these insights, we introduce class-wise *visual cues* as visual prototypes, following prototype learning frameworks [60]. As illustrated in Fig. 2, we mine visual cues from the fine-tuned visual features. Specifically, we gather all features belonging to each seen class and compute their mean. Formally, let $\mathcal{I}_c = \{\, i \mid y_i^s = c \,\}$ index the training samples of class $c \in \mathcal{Y}^s$, and let $\mathbf{x}_i^s \in \mathbb{R}^d$ denote the corresponding fine-tuned visual features. The visual prototype is:

$$\mathbf{v}^c \;=\; \frac{1}{|\mathcal{I}_c|}\sum_{i\in\mathcal{I}_c} \mathbf{x}_i^s \;\in\; \mathbb{R}^d. \tag{12}$$

During policy optimization, we impose a prototype-distillation loss $\mathcal{L}_{\text{PD}}$ that distills the synthesized features to align with the corresponding visual prototype:

$$\mathcal{L}_{\text{PD}} = \frac{1}{B}\sum_{i=1}^{B}\left(1 - \frac{\tilde{\mathbf{x}}_{0,i}^\top \mathbf{v}^{c_i}}{\|\tilde{\mathbf{x}}_{0,i}\|_2\, \|\mathbf{v}^{c_i}\|_2}\right). \tag{13}$$

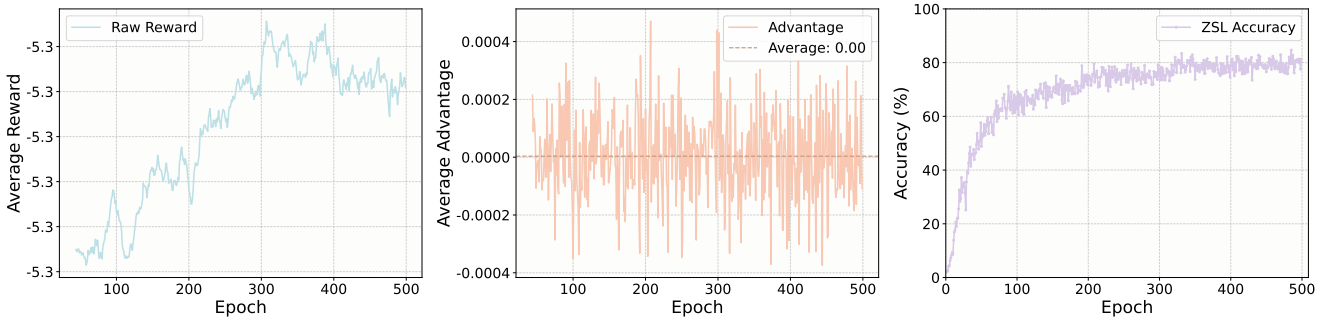


Figure 3. The training trends of our RLVC on CUB, including raw reward, EMA-adjusted advantage and ZSL accuracy.

This term pulls generated features closer to real data distribution and stabilizes the RL updates. We incorporate this loss into the generator update:

$$\mathcal{L}_G^{\text{total}} = \mathcal{L}_G^{\text{adv}} + \lambda_{\text{PD}} \mathcal{L}_{\text{PD}}, \tag{14}$$

where λ_{PD} is a coefficient. We optimize G_θ by minimizing $\mathcal{L}_G^{\text{total}}$. We also compare $\mathcal{L}_{\text{PD}}$ with alternative losses (*e.g.*, KL, ℓ_1) to validate the robustness of our design (Table 4).

3.4. Cold-Start Training Procedure

To ensure effective optimization, RLVC adopts a novel cold-start schedule inspired by post-training frameworks for large language models (*i.e.*, RL is disabled during the initial phase) [28]. Concretely, we first train for some epochs using the adversarial objective until the synthesized features exhibit basic class separability. Then, RL is activated to align generation with the downstream classification task further. To avoid gradient conflict after the cold-start threshold (*i.e.*, E_{RL}), we alternately update the policy model G_θ within each iteration via Eq. (14) and Eq. (11), rather than simply summing the two losses.

We summarize the complete procedure in Algorithm 1 for clarity. We also report the training trends of the raw reward, the EMA-adjusted advantage, and ZSL accuracy in Fig. 3. Empirically, the reward increases and then stabilizes, the advantage exhibits only small fluctuations, and the ZSL accuracy gains steadily. These trends consistently indicate that the model trains successfully and remains stable.

3.5. Inference for CZSL and GZSL

At inference time, we freeze the generator G_θ and synthesize visual features for unseen classes via Eq. (1). No additional tricks are used. For CZSL, we train a standard softmax classifier solely on the synthesized unseen features. For GZSL, we train the classifier on the union of the fine-tuned seen features and the synthesized unseen features. Further details of the classifier follow prior work [12, 33].

4. Experiments

In this section, we evaluate RLVC, aiming to answer the following questions: *(1) Can RLVC effectively and consistently improve ZSL accuracy on standard benchmarks (Tables 1 and 2)? (2) Do individual design choices of RLVC contribute to the observed accuracy gains (Tables 3 and 4)? (3) Does RLVC induce task-relevant feature distributions in the representation space (Fig. 4)? (4) How do hyperparameters impact performance (Fig. 5)?*

Algorithm 1 Training the Policy Model G_θ

Require: Minibatches $(\mathbf{x}, y, \mathbf{z}, \mathbf{v})$; weights λ_{PD}; RL start epoch E_{RL}; total epochs E; critic steps K.

Initialize G_θ, D_{x_0}, D_{x_t}; freeze reward model R.
procedure TRAINING(G_θ, D_{x_0}, D_{x_t})
  for epoch $= 1$ to E **do**
    for each minibatch **do**
      for $k = 1$ to K **do**
        Sample noise, timesteps $\tilde{\mathbf{x}}_0$ and $\tilde{\mathbf{x}}_t$.
        Update D_{x_0}, D_{x_t} via Eq. (4).
      end for
      Sample noise, timesteps, $\tilde{\mathbf{x}}_0$ and $\tilde{\mathbf{x}}_t$.
      Compute $\mathcal{L}_G^{\text{adv}}$ via Eq. (5).
      Compute $\mathcal{L}_{\text{PD}}$ via Eq. (13).
      Update G_θ via Eq.(14). ▷ adversarial
      if epoch $\geq E_{\text{RL}}$ **then** ▷ cold-start threshold
        Form advantage via Eqs. (6)–(10).
        Update G_θ via Eq. (11). ▷ RL update
      end if
    end for
  end for
  return optimized policy G_θ
end procedure

4.1. Experimental Setup

Datasets. We evaluate on three widely used ZSL datasets:

- **CUB** [74]: a fine-grained bird dataset with 11,788 images over 150/50 seen/unseen classes and 312 attributes.
- **SUN** [59]: a fine-grained scene corpus totals 14,340 images from 645/72 seen/unseen classes with 102 attributes.
- **AWA2** [76]: a coarse-grained animal dataset comprising 37,322 images across 40/10 seen/unseen classes with 85 attribute annotations.

Table 1. Compared our RLVC with the SOTA methods in CZSL and GZSL settings on CUB, SUN and AWA2 benchmarks. The symbol "⋆" indicates the semantic prototypes from the class name. The symbol "–" denotes that no results are provided in the original papers. The **bold** and underlined markings indicate the best and second-best results, respectively.

	Method	Backbone	CUB				SUN				AWA2			
			Acc	**U**	**S**	**H**	**Acc**	**U**	**S**	**H**	**Acc**	**U**	**S**	**H**
Embedding	CLIP ⋆ [ICML'21] [61]	ViT	–	55.2	54.8	55.0	–	–	–	–	–	–	–	–
	TransZero++ [TPAMI'22] [8]	ResNet	78.3	67.5	73.6	70.4	67.6	48.6	37.8	42.5	72.6	64.6	82.7	72.5
	DUET [AAAI'23] [18]	ViT	72.3	62.9	72.8	67.5	64.4	45.7	45.8	45.8	69.9	63.7	84.7	72.7
	ICIS [ICCV'23] [20]	ResNet	60.6	45.8	73.7	56.5	51.8	45.2	25.6	32.7	64.6	35.6	93.3	51.6
	HAS [ACM MM'23] [19]	ResNet	76.5	69.6	74.1	71.8	63.2	42.8	38.9	40.8	71.4	63.1	87.3	73.3
	I2DFormer+ [IJCV'24] [55]	ViT	45.9	38.3	55.2	45.3	–	–	–	–	77.3	69.8	83.2	75.9
	DSECN [CVPR'24] [47]	ResNet	40.9	–	–	45.3	40.0	–	–	38.5	49.1	–	–	53.7
	ZSLViT [CVPR'24] [14]	ViT	78.9	69.4	78.2	73.6	68.3	45.9	48.4	47.3	70.7	66.1	84.6	74.2
	PSVMA+ [TPAMI'24] [49]	ViT	78.8	71.8	77.8	74.6	74.5	61.5	49.4	54.8	79.2	74.2	86.4	79.8
	ZeroMamba [AAAI'25] [34]	VMamba	80.8	72.1	76.4	74.2	72.4	56.5	41.4	47.7	71.9	67.9	87.6	76.5
	AENet [AAAI'25] [50]	ViT	80.3	73.1	76.4	74.7	70.0	58.6	45.2	51.0	75.2	70.3	80.1	74.9
	VSPCN [CVPR'25] [37]	ViT	80.6	72.8	78.9	75.7	75.3	59.4	49.1	53.8	76.6	71.8	84.3	77.6
Generative	HSVA [NeurIPS'21] [7]	ResNet	–	52.7	58.3	55.3	–	48.6	39.0	43.3	–	56.7	79.8	66.3
	CE-GZSL [CVPR'21] [29]	ResNet	77.5	63.9	66.8	65.3	63.3	48.8	38.6	43.1	70.4	63.1	78.6	70.0
	SC-EGG [IJCAI'22] [32]	ResNet	75.1	64.1	73.6	68.5	69.2	45.1	43.6	44.3	78.2	60.9	89.3	72.4
	VGSE-APN [CVPR'22] [82]	ResNet	28.9	21.9	45.5	29.5	38.1	24.1	31.8	27.4	64.0	51.2	81.8	63.0
	ICCE [CVPR'22] [40]	ResNet	72.7	67.3	65.5	66.4	–	–	–	–	78.4	65.3	82.3	72.8
	FREE + ESZSL [ICLR'22] [4]	ResNet	–	51.6	60.4	55.7	–	48.2	36.5	41.5	–	51.3	78.0	61.8
	TDCSS [CVPR'22] [24]	ResNet	–	44.2	62.8	51.9	–	–	–	–	–	59.2	74.9	66.1
	DSP [ICML'23] [12]	ResNet	–	51.4	63.8	56.9	–	48.3	43.0	45.5	–	60.0	86.0	70.7
	EGANS [TEVC'23] [11]	ResNet	60.2	47.8	59.2	52.9	62.8	44.2	37.4	40.5	70.6	53.9	81.8	65.0
	ZeroNAS [TPAMI'23] [84]	ResNet	66.4	56.0	63.8	59.6	68.3	47.1	41.8	44.3	73.2	61.4	75.3	67.6
	CDL + OSCO [TPAMI'23] [3]	ResNet	–	29.0	69.0	40.6	–	32.0	**65.0**	42.9	–	48.0	71.0	57.1
	TF-VAEGAN + SHIP⋆ [ICCV'23][73]	ViT	–	21.1	**84.4**	34.0	–	–	–	–	–	43.7	**96.3**	60.1
	VADS [CVPR'24] [33]	ViT	86.8	74.1	74.6	74.3	76.3	**64.6**	49.0	55.7	82.5	75.4	83.6	79.3
	ViFR [IJCV'25] [17]	ResNet	74.5	63.9	72.0	67.6	69.2	51.3	40.0	44.7	77.8	68.2	78.9	73.2
	RLVC (ours)	ViT	**90.1**	**80.9**	81.4	**81.2**	**77.7**	59.6	55.6	**57.6**	**84.0**	**78.4**	82.4	**80.4**

Evaluation Metric. Following the standard evaluation protocol [13, 14], we evaluate RLVC under both CZSL and GZSL with average top-1 accuracy. In the CZSL setting, we report the accuracy on the test set of unseen classes (**Acc**). In the GZSL setting, we report accuracy on the seen (**S**) and unseen (**U**) test sets, together with their harmonic mean $\mathbf{H} = \mathbf{(2 \times S \times U) / (S + U)}$, which balances performance across the two splits.

Implementation Details. To ensure a fair comparison and reproducibility, we provide implementation details. For the visual encoder, we fine-tune a ViT to alleviate cross-dataset bias and use the `[CLS]` token to extract the visual feature [57]. We adopt the Adam optimizer (betas $= (0.5, 0.999)$) [38]. The learning rates are 5×10^{-4} for Eq. (5) and 5×10^{-5} for Eq. (11). We activate RL at $E_{\mathrm{RL}} = 30$ for CUB and SUN, and at $E_{\mathrm{RL}} = 7$ for AWA2. The prototype-distillation weight λ_{PD}, the number of synthesized samples per class, and the total number of training epochs are set to $\{20,\ 1,\ 5\}$, $\{400,\ 400,\ 4000\}$, and $\{500,\ 300,\ 30\}$ for CUB, SUN, and AWA2, respectively. All the experiments are run on a single NVIDIA RTX 4090 GPU (24 GB) and implemented using the PyTorch framework.

4.2. Core Results

Table 1 compares our RLVC with recent SOTA embedding and generative ZSL methods reported in top-tier venues. We highlight our findings below: **(1) RLVC exhibits the best CZSL accuracy on all three benchmarks**, *i.e.*, 90.1%, 77.7%, and 84.0% on CUB, SUN, and AWA2. These accuracies surpass the previous best method, VADS [33], by 3.1%, 1.4%, and 1.5%, respectively. **(2) In the GZSL setting, RLVC achieves the top harmonic mean H on all datasets and strikes a more effective trade-off between seen and unseen accuracy.** Compared with the second-best methods, like VSPCN [37] on CUB, VADS [33] on SUN, and PSVMA+ [49] on AWA2, RLVC notably increases **H** by 5.5%, 1.5%, and 0.6%, respectively. While RLVC does not achieve the best values of **S** and **U** on every dataset, it balances them more effectively, resulting in the highest **H** across all datasets. Both **U** and **S** are competitive against SOTA methods using ViT, VMamba, and ResNet backbones. Moreover, the VLM-based method SHIP [73] demonstrates outstanding performance on seen classes, yet it fails to generalize well to unseen classes.

Table 2. Effectiveness validation of RLVC across different semantic prototypes, including word embeddings of class names and expert-annotated attribute vectors. We mark the best results in **bold** and the accuracy gains (%) in parentheses.

Method	Semantic prototype	CUB				SUN				AWA2			
		Acc	U	S	H	Acc	U	S	H	Acc	U	S	H
Vanilla model	word embedding	61.8	52.4	66.3	58.5	72.0	55.4	46.4	50.5	73.1	62.8	82.3	71.2
RLVC	word embedding	**62.8** (+1.0)	51.5	70.9	**59.7** (+1.2)	**72.1** (+0.1)	55.0	50.3	**52.5** (+2.0)	**74.5** (+1.4)	61.3	91.7	**73.5** (+2.3)
Vanilla model	attribute vector	88.6	71.0	79.8	75.1	75.8	58.3	52.3	55.1	75.7	70.1	76.0	72.8
RLVC	attribute vector	**90.1** (+1.5)	80.9	81.4	**81.2** (+6.1)	**77.7** (+1.9)	59.6	55.6	**57.6** (+2.6)	**84.0** (+8.3)	78.4	82.4	**80.4** (+7.6)

Table 3. Results of RLVC variants on CUB, SUN and AWA2 datasets. We ablate specific components to assess their effectiveness. The **bold** marking indicates the best results.

Configuration	CUB		SUN		AWA2	
	Acc	H	Acc	H	Acc	H
RLVC w/o RL & visual cues	88.6	75.1	75.8	55.1	75.7	72.8
RLVC w/o RL (*i.e.*, Eq. (11))	89.2	80.1	76.1	55.6	79.4	73.9
RLVC w/o visual cues (*i.e.*, Eq. (13))	88.9	79.2	77.0	56.9	74.9	76.6
RLVC w/o fine-tuning visual encoder	89.2	77.5	76.0	56.4	81.1	76.3
RLVC w/o advantage (*i.e.*, Eq. (9))	89.6	79.7	76.0	56.0	82.4	78.2
RLVC	**90.1**	**81.2**	**77.7**	**57.6**	**84.0**	**80.4**

Table 4. Comparison results for different prototype-distillation losses combined with RLVC on CUB, SUN and AWA2 datasets. The **bold** marking indicates the best results.

Loss	CUB		SUN		AWA2	
	Acc	H	Acc	H	Acc	H
KL	88.9	80.1	77.2	**58.2**	76.4	76.4
ℓ_1	89.2	80.9	77.3	57.8	75.9	77.6
$\mathcal{L}_{\text{PD}}$	**90.1**	**81.2**	**77.7**	57.6	**84.0**	**80.4**

These results consistently demonstrate the superiority and strong generalization of our RLVC. Furthermore, they indicate that the proposed outcome-reward RL with visual cues effectively guides generation toward task-relevant features rather than merely distributionally plausible ones.

4.3. RLVC with Different Semantic Prototypes

To verify the reliability and show the robustness of our proposed RLVC, particularly for semantically similar classes, we conduct a comparative experiment and analysis across different semantic prototypes. We consider two types of prototypes: word embeddings of class names from the CLIP text encoder [61] and expert-annotated attribute vectors. Maintaining the same hyperparameters, we compare the performance of the full RLVC to that of a vanilla generative model (*i.e.*, without the proposed RL and visual cues).

As shown in Table 2, empirical results indicate that RLVC improves both **Acc** and **H** across all class prototype types and all datasets, with gains ranging from 0.1% to 8.3%. These results confirm that RLVC provides a faithful and task-relevant feature synthesis. Consequently, it enhances the discriminative power of visual features, even when using challenging word embeddings of class names that encode strong semantic similarity. To some extent, this accuracy increase also highlights the importance of aligning synthesized features with visual prototypes. It is worth noting that our RLVC also surpasses CLIP on CUB.

4.4. Ablation Study and Analysis

Ablation on Components. In Table 3, we conduct comprehensive ablation studies to evaluate the effectiveness of our proposed designs in RLVC. Applying our proposed operations results in a significant improvement in accuracy over the vanilla model. In CZSL, the average increase in **Acc** is 3.9%, and in GZSL, the average increase in **H** is 5.3%. **RL is crucial:** Without RL, the performance drops significantly, especially on the AWA2 dataset (*e.g.*, **Acc/H** drops from 84.0%/80.4% to 79.4%/73.9%). This indicates that RL significantly enhances the model's learning capabilities in both data distribution and task relevance. **Visual cues are beneficial:** Mining visual cues as visual prototypes and applying a prototype-distillation loss centralizes the synthesized features while also stabilizing training. It can be observed that omitting this component substantially degrades performance on all benchmarks. **Visual encoder fine-tuning is essential:** We do not optimize the visual encoder separately. Instead, we fine-tune it with the reward model. This reduces training overhead and simultaneously alleviates domain bias. The fine-tuned visual features inject dataset-specific priors, which is beneficial for GZSL (**H** increases by 3.7%, 1.2%, and 4.1% for CUB, SUN, and AWA2, respectively). **EMA reward smoothing outperforms raw reward:** Our investigation demonstrates that smoothing the reward using Eq. (9) offers substantial performance advantages over using the raw reward directly. This also suggests that algorithms specifically tailored for RL optimization are beneficial.

Ablation on Prototype-Distillation Losses. Compared to the standard KL loss, we introduce a novel prototype-distillation loss (Eq. (13)). As shown in Table 4, our loss outperforms KL and ℓ_1 in most cases, with the exception of the GZSL setting on SUN. We attribute this to the proposed loss being better suited for clustering and for distilling prototype information into the model.

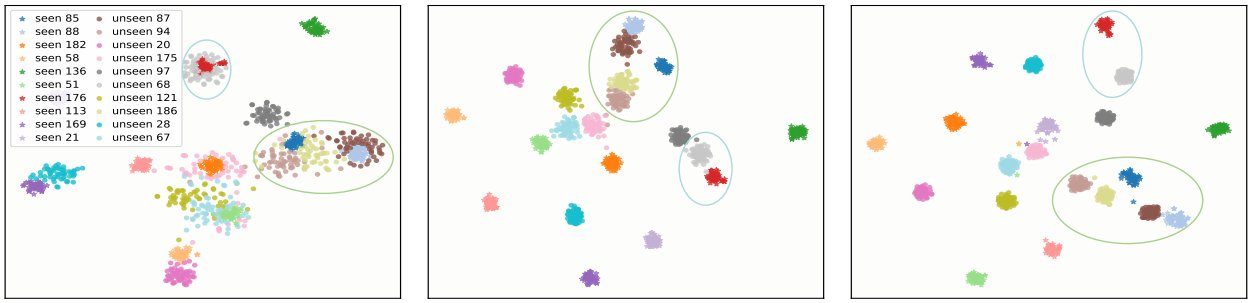


Figure 4. Qualitative t-SNE visualization of RLVC on CUB: (a) without RL and visual cues, (b) without visual cues, and (c) full RLVC. We use real features of seen classes and synthetic features of unseen classes. Zoom in for details.

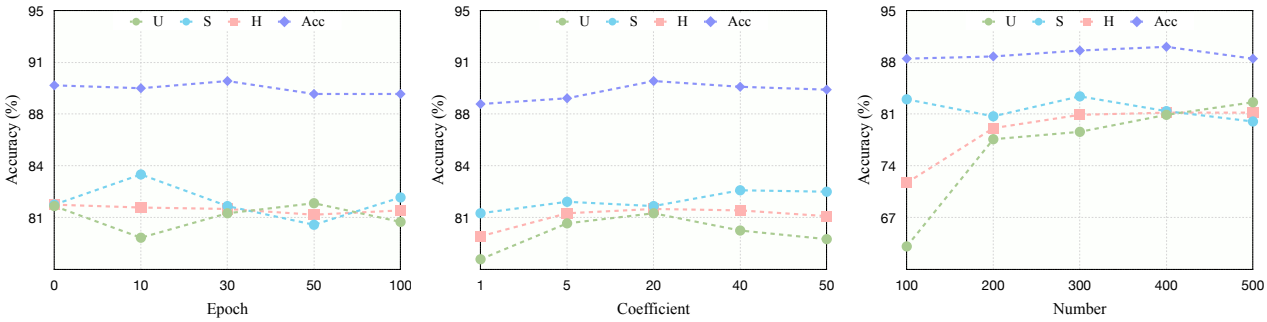


Figure 5. Effect of hyperparameters on CUB, including the epoch of RL cold-start, the coefficient of visual loss, and the number of synthetic unseen samples.

4.5. Qualitative Evaluation

To intuitively illustrate the distribution of the synthesized data, we use t-SNE visualization [68] to display the results from (a) the vanilla model (*i.e.*, without RL and visual cues), (b) the model without visual cues, and (c) our full RLVC on CUB in Fig. 4. The visualization includes 10 real seen classes and 10 synthesized unseen classes, denoted by $\star$ and $\circ$, respectively. The blue circle denotes "Baltimore Oriole" and "Orchard Oriole", and the green circle denotes "Harris Sparrow", "Lincoln Sparrow", "Le Conte Sparrow", "White crowned Sparrow" and "Tree Sparrow". Within each circle, they are semantically similar.

Visually, in (a) the vanilla model (*i.e.*, without RL and visual cues), seen and unseen classes are significantly overlapped, particularly for semantically similar categories. In contrast, (b) without visual cues and (c) our RLVC exhibit clear boundaries, contributing to classification. Comparatively, RLVC imposes a visual prototype constraint, leading to each class demonstrating more compact clustering. Overall, our RLVC learns a task-relevant, more accurate data distribution, which aligns with our motivation.

4.6. Hyperparameter Analysis

We conduct a sensitivity analysis for several key hyperparameters on the CUB to validate our default configuration, with results shown in Fig. 5. This analysis includes: the epoch of RL cold-start, the coefficient of visual loss, and the number of synthetic unseen samples. The cold-start mechanism stabilizes initial RL training. Our experiments show that a value of 30 ensures stable training while achieving optimal performance. The visual loss coefficient balances the generative objective against prototype-based feature clustering. As its value increases, performance first rises, then falls, peaking at 20. The number of synthetic samples for unseen classes impacts the accuracy balance between seen and unseen classes. The best **H** is achieved when it is set to 400. These results confirm the rationality and robustness of our hyperparameter choices. We provide the detailed settings for each dataset in §4.1.

5. Conclusion

In this work, we present RLVC, an outcome-reward reinforcement learning framework with visual cues for generative ZSL. This framework aims to learn features well-suited for classification by strengthening the model's generative capability. Equipped with a carefully designed cold-start training strategy, RLVC reliably synthesizes features. Through qualitative and quantitative experiments on three ZSL benchmarks, we show that RLVC consistently outperforms existing methods. As an initial exploration of reinforcement learning for generative ZSL, we highlight RLVC's potential. This approach may also benefit broader tasks and robust training recipes.

Acknowledgments

This work was supported by the National Key R&D Program of China (2023YFC3305600) and the National Natural Science Foundation of China (92570101).

References

[1] Mina Ghadimi Atigh, Stephanie Nargang, Martin Keller-Ressel, and Pascal Mettes. Simzsl: Zero-shot learning beyond a pre-defined semantic embedding space. *International Journal of Computer Vision*, pages 1–17, 2025. 2

[2] Jacopo Cavazza, Vittorio Murino, and Alessio Del Bue. No adversaries to zero-shot learning: Distilling an ensemble of gaussian feature generators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12167–12178, 2023. 2

[3] Jacopo Cavazza, Vittorio Murino, and Alessio Del Bue. No adversaries to zero-shot learning: Distilling an ensemble of gaussian feature generators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 6

[4] Samet Cetin. Closed-form sample probing for training generative models in zero-shot learning. Master's thesis, Middle East Technical University, 2022. 1, 2, 6

[5] Dubing Chen, Yuming Shen, Haofeng Zhang, and Philip HS Torr. Zero-shot logit adjustment. *arXiv preprint arXiv:2204.11822*, 2022. 2

[6] Shiming Chen, Wenjie Wang, Beihao Xia, Qinmu Peng, Xinge You, Feng Zheng, and Ling Shao. Free: Feature refinement for generalized zero-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 122–131, 2021. 1, 2

[7] Shiming Chen, GuoSen Xie, Yang Liu, Baigui Sun, Hao Li, Xinge You, and Ling Shao. Hsva: Hierarchical semantic-visual adaptation for zero-shot learning. *Advances in Neural Information Processing Systems*, 34:16622–16634, 2021. 2, 6

[8] Shiming Chen, Ziming Hong, Wenjin Hou, Guo-Sen Xie, Yibing Song, Jian Zhao, Xinge You, Shuicheng Yan, and Ling Shao. Transzero++: Cross attribute-guided transformer for zero-shot learning. *TPAMI*, 2022. 2, 6

[9] Shiming Chen, Ziming Hong, Yang Liu, Guo-Sen Xie, Baigui Sun, Hao Li, Qinmu Peng, Ke Lu, and Xinge You. Transzero: Attribute-guided transformer for zero-shot learning. In *AAAI*, page 3, 2022. 2

[10] Shiming Chen, Ziming Hong, Guo-Sen Xie, Wenhan Yang, Kai Wang, Jian Zhao, and Xinge You. Msdn: Mutually semantic distillation network for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7612–7621, 2022. 1

[11] Shiming Chen, Shuhuang Chen, Wenjin Hou, Weiping Ding, and Xinge You. Egans: Evolutionary generative adversarial network search for zero-shot learning. *IEEE Transactions on Evolutionary Computation*, 28(3):582–596, 2023. 2, 6

[12] Shiming Chen, Wenjin Hou, Ziming Hong, Xiaohan Ding, Yibing Song, Xinge You, Tongliang Liu, and Kun Zhang. Evolving semantic prototype improves generative zero-shot learning. *arXiv preprint arXiv:2306.06931*, 2023. 1, 2, 5, 6

[13] Shuhuang Chen, Dingjie Fu, Shiming Chen, Wenjin Hou, Xinge You, et al. Causal visual-semantic correlation for zero-shot learning. In *ACM Multimedia 2024*, 2024. 2, 6

[14] Shiming Chen, Wenjin Hou, Salman Khan, and Fahad Shahbaz Khan. Progressive semantic-guided vision transformer for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23964–23974, 2024. 2, 6

[15] Shuhuang Chen, Shiming Chen, Ziming Hong, Yuanjie Shao, and Xinge You. Dynamic semantic complementary network for zero-shot learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2025. 2

[16] Shiming Chen, Dingjie Fu, Salman Khan, and Fahad Shahbaz Khan. Genzsl: Generative zero-shot learning via inductive variational autoencoder. *arXiv preprint arXiv:2505.11882*, 2025. 1, 2

[17] Shiming Chen, Ziming Hong, Xinge You, and Ling Shao. Semantics-conditioned generative zero-shot learning via feature refinement. *International Journal of Computer Vision*, pages 1–18, 2025. 1, 2, 6

[18] Zhuo Chen, Yufeng Huang, Jiaoyan Chen, Yuxia Geng, Wen Zhang, Yin Fang, Jeff Z Pan, and Huajun Chen. Duet: Cross-modal semantic grounding for contrastive zero-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 405–413, 2023. 2, 6

[19] Zhi Chen, Pengfei Zhang, Jingjing Li, Sen Wang, and Zi Huang. Zero-shot learning by harnessing adversarial samples. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4138–4146, 2023. 2, 6

[20] Anders Christensen, Massimiliano Mancini, A Koepke, Ole Winther, and Zeynep Akata. Image-free classifier injection for zero-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19072–19081, 2023. 6

[21] Kevin Clark and Priyank Jaini. Text-to-image diffusion models are zero shot classifiers. *Advances in Neural Information Processing Systems*, 36:58921–58937, 2023. 1

[22] Bowen Duan, Shiming Chen, Yufei Guo, Guo-Sen Xie, Weiping Ding, and Yisong Wang. Visual–semantic graph matching net for zero-shot learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 2

[23] Rafael Felix, Ian Reid, Gustavo Carneiro, et al. Multi-modal cycle-consistent generalized zero-shot learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 21–37, 2018. 2

[24] Yaogong Feng, Xiaowen Huang, Pengbo Yang, Jian Yu, and Jitao Sang. Non-generative generalized zero-shot learning via task-correlated disentanglement and controllable samples synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9346–9355, 2022. 2, 6

[25] Dingjie Fu, Wenjin Hou, Shiming Chen, Shuhuang Chen, Xinge You, Salman Khan, and Fahad Shahbaz Khan. Discriminative image generation with diffusion models for zero-shot learning. *arXiv preprint arXiv:2412.17219*, 2024. 1, 2

[26] Majid Ghasemi, Amir Hossein Moosavi, and Dariush Ebrahimi. A comprehensive survey of reinforcement learn-

ing: From algorithms to practical challenges. *arXiv preprint arXiv:2411.18892*, 2024. 3

[27] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[28] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 3, 5

[29] Zongyan Han, Zhenyong Fu, Shuo Chen, and Jian Yang. Contrastive embedding for generalized zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2371–2381, 2021. 2, 6

[30] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[31] Hanbin Hong, Shenao Yan, Shuya Feng, Yan Yan, and Yuan Hong. Galot: Generative active learning via optimizable zero-shot text-to-image generation. *arXiv preprint arXiv:2412.16227*, 2024. 1

[32] Ziming Hong, Shiming Chen, G Xie, Wenhan Yang, Jian Zhao, Yuanjie Shao, Qinmu Peng, and Xinge You. Semantic compression embedding for generative zero-shot learning. *IJCAI, Vienna, Austria*, 7:956–963, 2022. 1, 2, 6

[33] Wenjin Hou, Shiming Chen, Shuhuang Chen, Ziming Hong, Yan Wang, Xuetao Feng, Salman Khan, Fahad Shahbaz Khan, and Xinge You. Visual-augmented dynamic semantic prototype for generative zero-shot learning. In *CVPR*, pages 23627–23637, 2024. 1, 2, 5, 6

[34] Wenjin Hou, Dingjie Fu, Kun Li, Shiming Chen, Hehe Fan, and Yi Yang. Zeromamba: Exploring visual state space model for zero-shot learning. *AAAI*, 2025. 1, 2, 6

[35] Dat Huynh and Ehsan Elhamifar. Fine-grained generalized zero-shot learning via dense attribute-based attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4483–4493, 2020. 2

[36] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. 3

[37] Huajie Jiang, Zhengxian Li, Xiaohan Yu, Yongli Hu, Baocai Yin, Jian Yang, and Yuankai Qi. Visual and semantic prompt collaboration for generalized zero-shot learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20275–20285, 2025. 2, 6

[38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 6

[39] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[40] Xia Kong, Zuodong Gao, Xiaofan Li, Ming Hong, Jun Liu, Chengjie Wang, Yuan Xie, and Yanyun Qu. En-compactness: Self-distillation embedding & contrastive generation for generalized zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9306–9315, 2022. 1, 2, 6

[41] Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. Llm post-training: A deep dive into reasoning large language models. *arXiv preprint arXiv:2502.21321*, 2025. 3

[42] Soyeong Kwon, Taegyeong Lee, and Taehwan Kim. Zero-shot text-guided infinite image synthesis with llm guidance. *arXiv preprint arXiv:2407.12642*, 2024. 1

[43] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *AAAI*, page 3, 2008. 2

[44] Jingjing Li, Mengmeng Jing, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. Leveraging the invariant side of generative zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7402–7411, 2019. 2

[45] Ming Li, Huazhu Fu, Shengfeng He, Hehe Fan, Jun Liu, Jussi Keppo, and Mike Zheng Shou. Dr-fer: Discriminative and robust representation learning for facial expression recognition. *IEEE Transactions on Multimedia*, 26:6297–6309, 2023. 1

[46] Xiaofan Li, Yachao Zhang, Shiran Bian, Yanyun Qu, Yuan Xie, Zhongchao Shi, and Jianping Fan. Vs-boost: Boosting visual-semantic association for generalized zero-shot learning. In *IJCAI*, pages 1107–1115, 2023. 1

[47] Yapeng Li, Yong Luo, Zengmao Wang, and Bo Du. Improving generalized zero-shot learning by exploring the diverse semantics from external class names. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23344–23353, 2024. 6

[48] Man Liu, Feng Li, Chunjie Zhang, Yunchao Wei, Huihui Bai, and Yao Zhao. Progressive semantic-visual mutual adaption for generalized zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15337–15346, 2023. 2

[49] Man Liu, Huihui Bai, Feng Li, Chunjie Zhang, Yunchao Wei, Meng Wang, Tat-Seng Chua, and Yao Zhao. Psvma+: Exploring multi-granularity semantic-visual adaption for generalized zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2, 6

[50] Man Liu, Huihui Bai, Feng Li, Chunjie Zhang, Yunchao Wei, Tat-Seng Chua, and Yao Zhao. Attend and enrich: Enhanced visual prompt for zero-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5504–5512, 2025. 2, 6

[51] Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025. 3

[52] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv:2503.01785*, 2025. 2, 3

[53] Muhammad Ferjad Naeem, Yongqin Xian, Luc V Gool, and Federico Tombari. I2dformer: Learning image to document attention for zero-shot image classification. *Advances in Neural Information Processing Systems*, 35:12283–12294, 2022. 1

[54] Muhammad Ferjad Naeem, Muhammad Gul Zain Ali Khan, Yongqin Xian, Muhammad Zeshan Afzal, Didier Stricker, Luc Van Gool, and Federico Tombari. I2mvformer: Large language model generated multi-view document supervision for zero-shot image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15169–15179, 2023. 2

[55] Muhammad Ferjad Naeem, Yongqin Xian, Luc Van Gool, and Federico Tombari. I2dformer+: Learning image to document summary attention for zero-shot image classification. *IJCV*, pages 1–17, 2024. 1, 2, 6

[56] Sanath Narayan, Akshita Gupta, Fahad Shahbaz Khan, Cees GM Snoek, and Ling Shao. Latent embedding feedback and discriminative features for zero-shot classification. In *European Conference on Computer Vision*, pages 479–495. Springer, 2020. 1, 2

[57] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6

[58] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. *Advances in neural information processing systems*, 22, 2009. 2

[59] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2751–2758. IEEE, 2012. 5

[60] Hongyu Qu, Jianan Wei, Xiangbo Shu, and Wenguan Wang. Learning clustering-based prototypes for compositional zero-shot learning. *arXiv:2502.06501*, 2025. 4

[61] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 6, 7

[62] Ashish Kumar Shakya, Gopinatha Pillai, and Sohom Chakrabarty. Reinforcement learning algorithms: A brief survey. *Expert Systems with Applications*, 231:120495, 2023. 2

[63] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 3

[64] Yufei Shi, Weilong Yan, Gang Xu, Yumeng Li, Yucheng Chen, Zhenxi Li, Fei Yu, Ming Li, and Si Yong Yeo. Pvchat: Personalized video chat with one-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23321–23331, 2025. 3

[65] Yueqi Song, Tianyue Ou, Yibo Kong, Zecheng Li, Graham Neubig, and Xiang Yue. Visualpuzzles: Decoupling multimodal reasoning evaluation from domain knowledge. *arXiv preprint arXiv:2504.10342*, 2025. 3

[66] Hongzu Su, Jingjing Li, Ke Lu, Lei Zhu, and Heng Tao Shen. Dual-aligned feature confusion alleviation for generalized zero-shot learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 33:3774–3785, 2023. 2

[67] Songbai Tan, Xuerui Qiu, Yao Shu, Gang Xu, Linrui Xu, Xiangyu Xu, Huiping Zhuang, Ming Li, and Fei Yu. WMarkGPT: Watermarked image understanding via multimodal large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 58621–58636. PMLR, 2025. 3

[68] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 8

[69] Maunil R Vyas, Hemanth Venkateswara, and Sethuraman Panchanathan. Leveraging seen and unseen semantic relationships for generative zero-shot learning. In *European Conference on Computer Vision*, pages 70–86. Springer, 2020. 2

[70] Chaoqun Wang, Shaobo Min, Xuejin Chen, Xiaoyan Sun, and Houqiang Li. Dual progressive prototype network for generalized zero-shot learning. *Advances in Neural Information Processing Systems*, 34:2936–2948, 2021. 2

[71] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 3

[72] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. 3

[73] Zhengbo Wang, Jian Liang, Ran He, Nan Xu, Zilei Wang, and Tieniu Tan. Improving zero-shot generalization for clip with synthesized prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3032–3042, 2023. 1, 6

[74] P. Welinder, S. Branson, T. Mita, C. Wah, Florian Schroff, Serge J. Belongie, and P. Perona. Caltech-ucsd birds 200. *Technical Report CNS-TR-2010-001, Caltech,*, 2010. 5

[75] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551, 2018. 1, 2

[76] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10275–10284, 2019. 1, 5

[77] Yongli Xiang, Ziming Hong, Zhaoqing Wang, Xiangyu Zhao, Bo Han, and Tongliang Liu. When safety collides: Resolving multi-category harmful conflicts in text-to-image diffusion via adaptive safety guidance. *arXiv preprint*, 2026. 2

[78] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021. 4

[79] Guo-Sen Xie, Li Liu, Xiaobo Jin, Fan Zhu, Zheng Zhang, Jie Qin, Yazhou Yao, and Ling Shao. Attentive region embedding network for zero-shot learning. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9384–9393, 2019. 2

[80] Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024. 3

[81] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele, and Zeynep Akata. Attribute prototype network for any-shot learning. *IJCV*, 130(7):1735–1753, 2022. 2

[82] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele, and Zeynep Akata. Vgse: Visually-grounded semantic embeddings for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9316–9325, 2022. 6

[83] Yi Xu, Chengzu Li, Han Zhou, Xingchen Wan, Caiqi Zhang, Anna Korhonen, and Ivan Vulić. Visual planning: Let's think only with images. *arXiv preprint arXiv:2505.11409*, 2025. 3

[84] Caixia Yan, Xiaojun Chang, Zhihui Li, Weili Guan, Zongyuan Ge, Lei Zhu, and Qinghua Zheng. Zeronas: Differentiable generative adversarial networks search for zero-shot learning. *IEEE transactions on pattern analysis and machine intelligence*, 2021. 1, 2, 6

[85] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 3

[86] Zihan Ye, Guanyu Yang, Xiaobo Jin, Youfa Liu, and Kaizhu Huang. Rebalanced zero-shot learning. *IEEE Transactions on Image Processing*, 32:4185–4198, 2023. 2

[87] Zihan Ye, Shreyank N Gowda, Shiming Chen, Xiaowei Huang, Haotian Xu, Fahad Shahbaz Khan, Yaochu Jin, Kaizhu Huang, and Xiaobo Jin. Zerodiff: Solidified visual-semantic correlation in zero-shot learning. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2, 4

[88] Bowen Zheng, Yongli Xiang, Ziming Hong, Zerong Lin, Chaojian Yu, Tongliang Liu, and Xinge You. Vii: Visual instruction injection for jailbreaking image-to-video generation models. *arXiv preprint*, 2026. 2