Read-Only Zero-Shot Classifier Expansion from Pairwise Semantics

Zhiqiang Zhong[1,2] **Jun Pang**[1,2]
[1]Institute for Advanced Studies, University of Luxembourg
[2]Faculty of Science, Technology and Medicine, University of Luxembourg
Contacts: zhiqiang.zhong@uni.lu

Abstract

Image-free Zero-Shot Learning (I-ZSL) aims to extend a pre-trained classifier to unseen classes without using training images or image features during adaptation. Existing I-ZSL methods often depend on pre-defined class descriptions and fixed text encoders, which can be misaligned with the classifier space of the deployed model. We propose *Geometry-Regularized Relational Weight Synthesis* (GEORWS), a read-only classifier expansion framework that learns to synthesize unseen-class classifier weights from pairwise semantic relations and observed seen-class weights. Given class-pair affinity descriptions generated by a frozen large language model, GEORWS uses an adaptive semantic encoder to estimate affinity coefficients over seen classifiers through leave-one-out reconstruction. The training objective further includes a classifier-geometry regularizer, which anchors the learned coefficients to neighborhoods in the seen-class classifier head, and a semantic-consistency regularizer, which improves stability under perturbations of pairwise semantic descriptions. At inference, GEORWS synthesizes unseen-class weights as mixtures of seen-class weights and injects them into the classifier head while keeping the feature extractor fixed. Experiments on standard ZSL and GZSL benchmarks show consistent improvements over image-free and adapted zero-shot baselines, especially on fine-grained recognition tasks.

1 Introduction

AI models have achieved strong performance in visual recognition by learning rich representations from large-scale datasets [18, 17]. However, the ability to access, share, and reuse image data is often limited by privacy policies, copyright restrictions, and security constraints [23]. These restrictions are common in sensitive domains where data access is tightly regulated, and in industrial settings where datasets are proprietary and cannot be shared across organizations. As a result, when a deployed classifier must be extended to new labels, it is often infeasible to retrain or adapt the model using the original training images. This motivates a read-only classifier expansion problem: *given a pre-trained classifier and its seen-class classifier head, can we add unseen classes without using training images or image features during adaptation?*

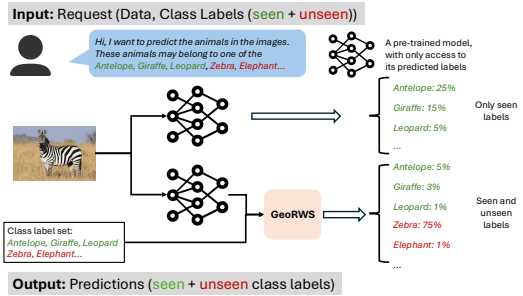

Figure 1: Overview of GEORWS. Expansion uses read-only access to seen-class classifier weights and class labels, with no images or image features. The test image is used only for final prediction.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Zero-Shot Learning (ZSL) addresses the recognition of unseen classes by using semantic side information, such as class names, attributes, or textual descriptions, to connect seen and unseen categories [29]. *Generalized Zero-Shot Learning* (GZSL) further evaluates the more realistic case where test instances may come from both seen and unseen classes, requiring a balance between unseen-class transfer and seen-class retention. Although ZSL and GZSL reduce the need for labeled examples from every class, most existing methods still assume access to training images or image features during training or adaptation, or require visual models to extract such features [32, 5]. These assumptions do not hold in image-free settings where the adaptation process must rely only on the deployed classifier and semantic label information.

Image-free Zero-Shot Learning (I-ZSL) formalizes this stricter setting [6]. Given a pre-trained classifier and a target label set, I-ZSL aims to extend the classifier to unseen classes without using training images during the extension process. ICIS [6] learns mappings between class semantics and classifier weights using class names or attributes. Class-wise semantic inputs nevertheless have two practical limits. First, pre-defined class descriptions assume that accurate semantic annotations are available, which can be difficult for fine-grained categories where class differences are subtle and domain-dependent [33]. Second, raw similarities from fixed text encoders need not match neighborhoods in the classifier head [34], motivating explicit classifier-geometry control. These limitations suggest that I-ZSL methods should use task-relevant class relations and should align semantic coefficients with the classifier geometry of the deployed model.

In this work, we propose *Geometry-Regularized Relational Weight Synthesis* (GEORWS), a framework for read-only zero-shot classifier expansion from pairwise semantics. As illustrated in Figure 1, GEORWS uses a frozen *Large Language Model* (LLM) to generate *class-pair affinity descriptions* that describe shared properties and key differences between classes. Instead of treating these descriptions as direct class prototypes, GEORWS maps them through an adaptive semantic encoder and estimates affinity coefficients over seen-class classifier weights. The encoder and scoring head are trained by leave-one-out reconstruction on seen classes: one seen-class weight is masked and reconstructed as a mixture of the remaining seen-class weights using pairwise semantic descriptions. This training rule directly matches the inference-time goal of synthesizing unseen-class classifier weights from seen-class weights. To improve the stability of this synthesis process, GEORWS adds two training-time regularizers. A classifier-geometry regularizer anchors the learned affinity coefficients to local neighborhoods in the seen-class classifier head, using only observed seen-class weights. A semantic-consistency regularizer further encourages the synthesized weights and affinity coefficients to remain stable under perturbations of the pairwise semantic embeddings. After training, GEORWS synthesizes each unseen-class classifier weight as a mixture of seen-class weights and injects the synthesized weights into the classifier head. The feature extractor is kept fixed throughout adaptation, and no training images or image features are used to construct the unseen-class weights.

We evaluate GEORWS on benchmarks under the I-ZSL protocol. The results show consistent improvements over image-free and adapted zero-shot baselines, with strong gains on fine-grained recognition tasks where pairwise class relations are especially useful. Further analyses validate the roles of pairwise affinity descriptions, adaptive semantic encoding, classifier-geometry regularization, and semantic-consistency regularization. These results support GEORWS as a practical framework for extending a deployed classifier to new labels when image data cannot be used during adaptation.

2 Related Work

Zero-shot learning. *Zero-Shot Learning* (ZSL) aims to recognize classes without labeled training examples by transferring knowledge from seen to unseen categories through semantic side information [22, 21, 29]. Early methods build semantic spaces from attributes, word embeddings, or class descriptions and align visual representations with these semantic spaces [11, 16, 31, 37, 30, 35]. Recent ZSL and GZSL methods improve this transfer through stronger semantic prototypes, cross-modal alignment, generative modeling, or vision-language models [25, 4, 15, 32, 5, 26]. However, most of these methods require access to training images, extracted image features, or trainable vision components during adaptation. This makes them less suitable for settings where the deployed classifier must be extended without reusing the original image data.

Image-free zero-shot learning. *Image-free Zero-Shot Learning* (I-ZSL) studies a stricter setting in which a pre-trained classifier is extended to unseen classes without using training images during

Table 1: Comparison of GEORWS and related methods.

MODEL	I-ZSL COMPATIBLE	NO HUMAN-WRITTEN CLASS DESCRIPTIONS	TRAINABLE SEMANTIC-TO CLASSIFIER ALIGNMENT	NO IMAGE FEATURES DURING ADAPTATION
CONSE [22]	✔	✘	✘	✔
COSTA [21]	✔	✘	✘	✔
SUB.REG. [1]	✘	✔	✘	✘
WDAE [13]	✘	✔	✘	✘
WAVG [38]	✘	✘	✔	✘
SMO [38]	✘	✘	✔	✘
ICIS [6]	✔	✔	✔	✔
LABO [39]	✘	✔	✔	✘
ZSLVIT [5]	✘	✘	✔	✘
COMC [19]	✘	✔	✔	✘
DFZSL [32]	✘	✘	✔	✘
FUDD [10]	✘	✘	✔	✘
GENZSL [3]	✘	✔	✔	✘
GEORWS (OURS)	✔	✔	✔	✔

the extension process [6]. ICIS [6] is a representative I-ZSL method. It learns mappings between class-wise semantic representations and classifier weights, using attributes or encoded class names as semantic inputs. This design avoids image access during adaptation, but class-wise semantic inputs have two practical limits. First, they do not explicitly describe target–reference differences, which can matter for fine-grained categories [33]. Second, similarity in the frozen text embedding space need not reflect local neighborhoods in the deployed classifier head [34]. GEORWS addresses these issues by using pairwise class relations and by training the semantic-to-affinity mapping with classifier-space supervision from the seen-class weights.

Large language models for semantic representations. *Large Language Models* (LLMs) can generate rich semantic descriptions from class names and have been used in zero-shot and cross-modal learning [2, 27, 38, 32]. Menon and Vondrick [20] generate class-wise visual descriptions and classify test images by matching their CLIP features to these descriptions. GEORWS instead learns target–reference relations from seen classifier weights to synthesize unseen weights without images or image features during expansion. For I-ZSL, LLMs are useful because they can produce semantic information for new label sets without manual annotation. However, LLM-generated descriptions may be noisy, incomplete, or inconsistent across class pairs, and their text representations are not automatically aligned with the classifier space of a deployed visual model. Therefore, using LLM outputs directly as class prototypes can be unreliable. GEORWS instead treats LLM-generated class-pair descriptions as semantic side information and learns affinity coefficients through leave-one-out reconstruction on seen-class classifier weights. The classifier-geometry regularizer further anchors the coefficients to local neighborhoods in the classifier head, while the semantic-consistency regularizer improves stability under perturbations of pairwise descriptions.

Discussion. Table 1 compares GEORWS with related methods along four dimensions: compatibility with the I-ZSL protocol, dependence on human-written class descriptions, use of trainable semantic-to-classifier alignment, and use of image features during adaptation. Compared with ICIS, GEORWS uses automatically generated class-pair relations instead of class-wise semantic representations, and regularizes the learned coefficients using classifier geometry and semantic consistency. Compared with standard ZSL methods adapted to the image-free setting, GEORWS is designed directly for read-only classifier expansion: it uses no training images or image features during adaptation, keeps the feature extractor fixed, and synthesizes unseen-class weights from observed seen-class weights. These design choices support a more stable mapping from pairwise semantics to classifier weights, especially in fine-grained recognition settings. A more detailed discussion is provided in Appendix D.

3 Proposed Framework: GEORWS

3.1 Preliminary

Let $\Phi : \mathbf{X} \to \mathcal{S}$ be a pre-trained image classification model, where $\mathbf{X}$ is the input image space and $\mathcal{S}$ is the set of *seen* class labels. We model Φ as a composition of a feature extractor and a classifier head. Given an input image $x \in \mathbf{X}$, the feature extractor $\mathcal{F} : \mathbf{X} \to \mathbb{R}^D$ maps x to a D-dimensional representation, and the classifier head produces class scores $f_s(x) = \mathbf{w}_s^\top \mathcal{F}(x)$ for $s \in \mathcal{S}$, where $\mathbf{w}_s$

denotes the classifier weight vector associated with class s. We collect the classifier weights for seen classes as $\mathbf{W}_{\mathcal{S}} = [\mathbf{w}_s]_{s\in\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|\times D}$. The predicted label is given by $\Phi(x) = \arg\max_{s\in\mathcal{S}} f_s(x)$.

Image-free Zero-Shot Learning (I-ZSL) aims to extend Φ to a target label set $\mathcal{Y}$ without using training images or image features during adaptation. We denote the extended classification model by $\hat{\Phi} : \mathbf{X} \to \mathcal{Y}$ and consider two evaluation settings: (*i*) *Zero-Shot Learning* (ZSL), where $\mathcal{Y} \cap \mathcal{S} = \emptyset$ and all target classes are unseen, and (*ii*) *Generalized Zero-Shot Learning* (GZSL), where $\mathcal{Y} = \mathcal{S} \cup \mathcal{U}$ includes both seen and unseen classes. The unseen class set is $\mathcal{U} = \mathcal{Y} \setminus \mathcal{S}$.

During adaptation, the feature extractor $\mathcal{F}$ is kept fixed and is not trained. The seen-class classifier weights $\mathbf{W}_{\mathcal{S}}$ are accessible from the classifier head and provide the only classifier-side supervision. At test time, the deployed feature extractor remains callable for prediction with the extended classifier head. To support recognition of unseen classes without training image data, I-ZSL methods rely on semantic side information for classes in $\mathcal{S}$ and $\mathcal{U}$, such as class names, attributes, or textual descriptions. In existing approaches, this side information is often static and manually designed, while the text encoders used to process it are trained on general-purpose corpora. Thus, the induced semantic space may fail to capture task-specific, fine-grained inter-class differences and may be misaligned with the classifier space of Φ, leading to degraded performance in specialized domains [34].

3.2 Overview of the Framework

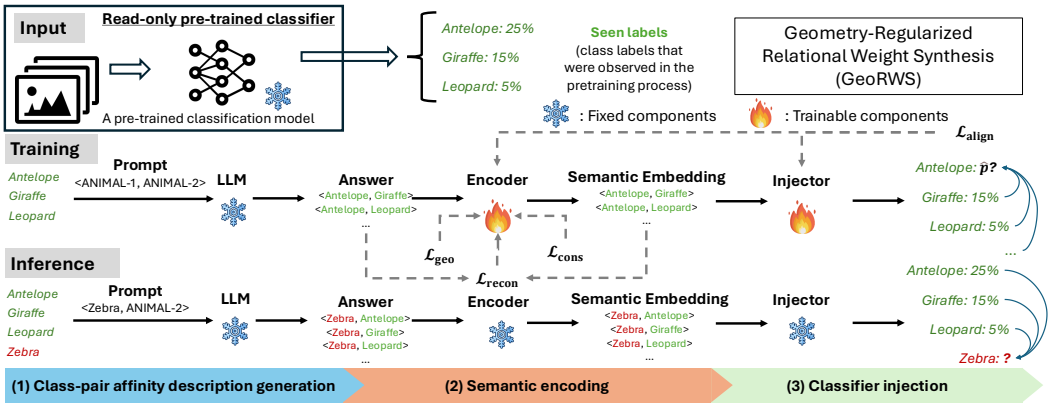

Figure 2: Overview of GeoRWS. Cached pairwise descriptions and seen-class weights determine synthesis coefficients. Leave-one-out training reconstructs masked seen weights; inference applies the learned rule to unseen classes. The semantic encoder, decoder, and scorer are trainable; the LLM, text encoder, backbone, and seen weights are frozen. Appendix C gives a concrete training and inference example.

GeoRWS extends a pre-trained image classification model Φ to recognize unseen classes under the I-ZSL setting. Unlike prior approaches that rely on static, manually designed class descriptions, GeoRWS uses class-pair semantic relationships to infer classifier weights for unseen classes without access to training images. As illustrated in Figure 2, GeoRWS consists of three sequential components. (*i*) A *class-pair affinity description generation module* uses a frozen Large Language Model (LLM) to produce task-aware descriptions that characterize semantic relationships between classes. (*ii*) An *adaptive semantic encoder* maps these descriptions into representations that are compatible with the classifier space of Φ. (*iii*) A *classifier injection module* synthesizes unseen-class classifier weights from seen-class weights and injects them into an extended model $\hat{\Phi}$ for prediction over $\mathcal{Y}$.

The trainable modules are optimized by leave-one-out reconstruction on seen classes. To reduce coefficient instability caused by noisy semantic descriptions, we further regularize the learned affinity coefficients with the local geometry of the classifier head and enforce consistency under semantic perturbations. Overall, GeoRWS transforms class-pair semantic relationships into classifier weights for unseen classes, enabling image-free extension of Φ while keeping the feature extractor fixed.

3.3 Class-pair Affinity Description

Class-pair affinity descriptions encode how two class labels relate to each other and provide the semantic signal used to compute class affinities for classifier injection. Instead of relying on fixed class attributes or descriptions, GEORWS uses a frozen LLM to generate *pairwise* descriptions tailored to the target label set and the classification task. We use the same description procedure for seen–seen pairs during leave-one-out training and unseen–seen pairs during inference.

For a target-reference pair (c, s), where $c = m \in \mathcal{S}$ during training and $c = u \in \mathcal{U}$ during inference, we query the LLM to produce a short, structured description that highlights shared properties and key differences between the two classes. The prompt template is given in Appendix A.1. For instance, on CUB, the relationship between a *sparrow* and a *warbler* may be described as: "Both are small songbirds with similar feeding habits, but they differ in plumage coloration." Compared with class-level text, these pairwise descriptions explicitly target inter-class distinctions, which is important for transferring knowledge from seen to unseen classes. We convert each textual description into a vector using a frozen text encoder. Let $f_{\text{LLM}}(c, s)$ denote the generated description and f_{ENC} denote the text encoder. The resulting affinity embedding is $\mathbf{a}_{c,s} = f_{\text{ENC}}\big(f_{\text{LLM}}(c, s)\big)$, where $\mathbf{a}_{c,s} \in \mathbb{R}^d$. The collection of pairwise affinity embeddings is consumed by the adaptive semantic encoder in Section 3.4 to produce representations aligned with the classifier space of Φ.

3.4 Adaptive Semantic Encoding for Classifier Injection

The adaptive semantic encoder maps each class-pair affinity embedding into a representation that is useful for composing classifier weights from the seen-class weights of Φ. Under I-ZSL, training images and image features are unavailable during adaptation, so we treat the seen-class classifier weights $\mathbf{W}_{\mathcal{S}} = \{\mathbf{w}_s\}_{s\in\mathcal{S}}$ as the only available proxy for the classifier space of Φ. Our goal is to learn a semantic-to-affinity mapping that assigns each target class a set of coefficients over seen classes, which can be used to synthesize its classifier weight.

Semantic embedding. Given a pairwise affinity embedding $\mathbf{a}_{c,s}$, a trainable encoder f_θ produces a representation $\mathbf{z}_{c,s} = f_\theta(\mathbf{a}_{c,s}) \in \mathbb{R}^k$. To score the relevance of a seen class s as a reference for synthesizing the classifier of target class c, we condition the score on both the semantic signal and the seen-class prototype in classifier space: $g_{c,s} = f_\psi\big(\mathbf{z}_{c,s} \oplus \mathbf{w}_s\big)$, where f_ψ is a trainable scoring head implemented as an MLP [12], and $\oplus$ denotes concatenation. For an unseen class u, we normalize $\{g_{u,s}\}_{s\in\mathcal{S}}$ with a softmax: $\alpha_{u,s} = \frac{\exp(g_{u,s})}{\sum_{s'\in\mathcal{S}}\exp(g_{u,s'})}, \quad s \in \mathcal{S}$.

Using the normalized coefficients, we synthesize the unseen-class classifier weight as $\mathbf{w}_u = \sum_{s\in\mathcal{S}} \alpha_{u,s}\mathbf{w}_s$. Because the coefficients are nonnegative and sum to one, the synthesized classifier lies in the convex hull of the seen-class classifier weights. This constraint keeps the injected weights within the observed classifier-space geometry while allowing the mixture coefficients to be guided by task-aware pairwise semantics. The same scoring rule is used during leave-one-out training by replacing the unseen class u with a masked seen class m and normalizing over $\mathcal{S} \setminus \{m\}$.

Classifier injection. Finally, we form the extended classifier by augmenting the original weight set with the synthesized unseen weights:

$$\mathbf{W}_{\mathcal{Y}} = \mathbf{W}_{\mathcal{S}} \cup \mathbf{W}_{\mathcal{U}}, \qquad \mathbf{W}_{\mathcal{U}} = \{\mathbf{w}_u\}_{u\in\mathcal{U}}. \tag{1}$$

The resulting model $\hat{\Phi} : \mathbf{X} \to \mathcal{Y}$ uses the fixed feature extractor of Φ and the augmented classifier weights for prediction over $\mathcal{Y}$.

3.5 Training and Inference

Training. We train the adaptive semantic encoder f_θ, the decoder $f_{\tilde{\theta}}$, and the scoring head f_ψ by leave-one-out reconstruction on seen classes. For each masked seen class $m \in \mathcal{S}$ and each reference class $s \in \mathcal{S} \setminus \{m\}$, we first compute the fixed pairwise affinity embedding $\mathbf{a}_{m,s} = f_{\text{ENC}}\big(f_{\text{LLM}}(m, s)\big)$. The trainable modules then produce $\mathbf{z}_{m,s} = f_\theta(\mathbf{a}_{m,s}), \quad g_{m,s} = f_\psi\big(\mathbf{z}_{m,s} \oplus \mathbf{w}_s\big)$, where $\mathbf{w}_s$ conditions affinity estimation on the classifier-space prototype of class s. The logits are normalized over the remaining seen classes: $\alpha_{m,s} = \frac{\exp(g_{m,s})}{\sum_{s'\in\mathcal{S}\setminus\{m\}}\exp(g_{m,s'})}, \quad s \in \mathcal{S} \setminus \{m\}$. The masked classifier weight is reconstructed by $\hat{\mathbf{w}}_m = \sum_{s\in\mathcal{S}\setminus\{m\}} \alpha_{m,s}\mathbf{w}_s$. We use the alignment loss

$\mathcal{L}_{\text{align}} = \sum_{m\in\mathcal{S}} \|\hat{\mathbf{w}}_m - \mathbf{w}_m\|_2^2$ to match the reconstructed weight to the original seen-class weight. We also retain the semantic reconstruction loss

$$\mathcal{L}_{\text{recon}} = \sum_{m\in\mathcal{S}} \sum_{s\in\mathcal{S}\setminus\{m\}} \left\|\mathbf{a}_{m,s} - f_{\tilde{\theta}}(\mathbf{z}_{m,s})\right\|_2^2,$$

which prevents the adaptive encoder from discarding information in the LLM-derived embeddings.

To regularize the learned coefficients, we introduce a classifier-geometry teacher distribution computed only from seen-class weights: $p^W_{m,s} = \frac{\exp(\cos(\mathbf{w}_m,\mathbf{w}_s)/\tau_W)}{\sum_{s'\in\mathcal{S}\setminus\{m\}}\exp(\cos(\mathbf{w}_m,\mathbf{w}_{s'})/\tau_W)}, \quad s \in \mathcal{S}\setminus\{m\}$, where τ_W is a temperature. The classifier-geometry loss is

$$\mathcal{L}_{\text{geo}} = \sum_{m\in\mathcal{S}} \text{KL}\left(p^W_{m,\cdot} \,\|\, \alpha_{m,\cdot}\right).$$

It encourages the semantic affinity coefficients to agree with local neighborhoods in classifier head.

To reduce sensitivity to noisy semantic descriptions, we further impose consistency under semantic perturbation. Let $\tilde{\mathbf{a}}_{m,s} = \text{Aug}(\mathbf{a}_{m,s})$ be an augmented affinity embedding. In the main experiments, $\text{Aug}(\cdot)$ is implemented as embedding dropout applied to $\mathbf{a}_{m,s}$ during training. We use description-level perturbations, such as truncation and paraphrasing, only for robustness analysis. Using f_θ and f_ψ, we compute $\tilde{\alpha}_{m,s}$ and $\tilde{\mathbf{w}}_m$ from $\tilde{\mathbf{a}}_{m,s}$ in the same way as above. The consistency loss is

$$\mathcal{L}_{\text{cons}} = \sum_{m\in\mathcal{S}} \|\hat{\mathbf{w}}_m - \tilde{\mathbf{w}}_m\|_2^2 + \sum_{m\in\mathcal{S}} D_{\text{SKL}}\left(\alpha_{m,\cdot}, \tilde{\alpha}_{m,\cdot}\right),$$

where $D_{\text{SKL}}(p,q) = \text{KL}(p\|q) + \text{KL}(q\|p)$. The full objective is

$$\mathcal{L} = \mathcal{L}_{\text{align}} + \lambda\mathcal{L}_{\text{recon}} + \beta\mathcal{L}_{\text{geo}} + \gamma\mathcal{L}_{\text{cons}}, \tag{2}$$

where λ, β, and γ control semantic reconstruction, classifier-geometry regularization, and semantic consistency, respectively. We optimize $(\theta, \tilde{\theta}, \psi)$ for T steps as summarized in Algorithm 1.

Inference. After training, we synthesize a classifier weight for each unseen class $u \in \mathcal{U}$ by relating it to all seen classes. For each $s \in \mathcal{S}$, we retrieve the cached embedding $\mathbf{a}_{u,s}$ and compute the learned representation and score:

$$\mathbf{a}_{u,s} = f_{\text{ENC}}\big(f_{\text{LLM}}(u,s)\big), \quad \mathbf{z}_{u,s} = f_\theta(\mathbf{a}_{u,s}), \quad g_{u,s} = f_\psi(\mathbf{z}_{u,s} \oplus \mathbf{w}_s).$$

The coefficients over seen classes are $\alpha_{u,s} = \frac{\exp(g_{u,s})}{\sum_{s'\in\mathcal{S}}\exp(g_{u,s'})}, \quad s \in \mathcal{S}$. The unseen-class classifier weight is synthesized as $\mathbf{w}_u = \sum_{s\in\mathcal{S}} \alpha_{u,s}\mathbf{w}_s$. All synthesized unseen weights are injected into the classifier head: $\mathbf{W}_\mathcal{Y} = \mathbf{W}_\mathcal{S} \cup \mathbf{W}_\mathcal{U}, \quad \mathbf{W}_\mathcal{U} = \{\mathbf{w}_u\}_{u\in\mathcal{U}}$. Given a test image x, prediction uses cosine scores: $\hat{\Phi}(x) = \arg\max_{y\in\mathcal{Y}} N(\mathbf{w}_y)^\top N(\mathcal{F}(x))$, where $N(\mathbf{v}) = \mathbf{v}/\|\mathbf{v}\|_2$ for $\mathbf{v} \neq \mathbf{0}$ and $N(\mathbf{0}) = \mathbf{0}$. Normalization is applied separately to every seen and unseen classifier weight at prediction; synthesis and reconstruction use the original weights. The stored parameters remain unchanged, although this scoring rule can change the original linear classifier's predictions. No training images or image features are used to construct $\mathbf{W}_\mathcal{U}$.

3.6 Theoretical Analysis

We provide a compact analysis of the classifier synthesis rule and the two training regularizers. The following analysis concerns the unnormalized synthesized weights. All proofs are given in Appendix E. Since the synthesized classifier is $\mathbf{w}_u = \sum_{s\in\mathcal{S}} \alpha_{u,s}\mathbf{w}_s$ with softmax coefficients, it lies in the convex hull $\mathcal{C}_\mathcal{S} = \text{conv}(\{\mathbf{w}_s : s \in \mathcal{S}\})$.

Proposition 1 (Approximation–estimation decomposition)**.** *Let $\mathbf{w}_u^\dagger$ denote the ideal classifier weight for unseen class u under the fixed feature extractor $\mathcal{F}$. Let $\mathbf{w}_u^\circ = \arg\min_{\mathbf{w}\in\mathcal{C}_\mathcal{S}} \|\mathbf{w} - \mathbf{w}_u^\dagger\|_2$ be the best approximation to $\mathbf{w}_u^\dagger$ inside the seen-weight convex hull. Then the synthesized classifier satisfies*

$$\|\mathbf{w}_u - \mathbf{w}_u^\dagger\|_2 \leq \underbrace{\|\mathbf{w}_u - \mathbf{w}_u^\circ\|_2}_{\textit{coefficient estimation error}} + \underbrace{\text{dist}(\mathbf{w}_u^\dagger, \mathcal{C}_\mathcal{S})}_{\textit{convex-hull approximation error}}. \tag{3}$$

Table 2: Comparison between our GEORWS framework and baselines using standard benchmarks (*i.e.*, CUB, AWA2, and SUN). We measure the results as unseen accuracy (Acc) for the zero-shot task, unseen (u) and seen (s) accuracy and their harmonic mean (H) for the generalized zero-shot setting. All results report mean ± standard deviation over five matched runs.

Models	Zero-Shot Accuracy			Generalized Zero-Shot Accuracy								
	CUB	AWA2	SUN	CUB			AWA2			SUN		
	Acc	Acc	Acc	u	s	H	u	s	H	u	s	H
ConSE	$41.39_{\pm 0.79}$	$44.94_{\pm 0.52}$	$43.77_{\pm 0.39}$	$0.45_{\pm 0.03}$	**87.83**$_{\pm 0.87}$	$0.90_{\pm 0.06}$	$3.22_{\pm 0.13}$	**96.29**$_{\pm 0.15}$	$6.23_{\pm 0.24}$	$0.09_{\pm 0.00}$	$49.43_{\pm 0.27}$	$0.18_{\pm 0.00}$
COSTA	$33.62_{\pm 1.21}$	$46.30_{\pm 1.63}$	$18.68_{\pm 0.97}$	$0.00_{\pm 0.00}$	$87.82_{\pm 1.70}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$96.27_{\pm 1.72}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	**51.90**$_{\pm 2.04}$	$0.00_{\pm 0.00}$
Sub.Reg.*	$51.84_{\pm 1.58}$	$46.06_{\pm 1.14}$	$44.38_{\pm 1.28}$	$1.17_{\pm 0.41}$	**87.83**$_{\pm 1.98}$	$2.31_{\pm 0.80}$	$0.00_{\pm 0.00}$	$96.26_{\pm 1.25}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$51.55_{\pm 1.50}$	$0.00_{\pm 0.00}$
wDAE*	$48.88_{\pm 0.90}$	$52.47_{\pm 1.30}$	$45.76_{\pm 2.53}$	$0.93_{\pm 0.05}$	$87.45_{\pm 1.44}$	$1.84_{\pm 0.10}$	$0.00_{\pm 0.00}$	$95.86_{\pm 2.17}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$50.43_{\pm 1.04}$	$0.00_{\pm 0.00}$
WAvg*	$2.00_{\pm 0.43}$	$20.42_{\pm 0.61}$	$1.39_{\pm 0.53}$	$1.75_{\pm 0.20}$	$58.97_{\pm 2.67}$	$3.40_{\pm 0.38}$	$8.77_{\pm 0.10}$	$87.05_{\pm 2.06}$	$15.93_{\pm 0.17}$	$1.34_{\pm 0.35}$	$7.57_{\pm 0.90}$	$2.28_{\pm 0.51}$
SMO*	$45.52_{\pm 2.48}$	$55.67_{\pm 5.50}$	$43.68_{\pm 2.17}$	$40.11_{\pm 1.48}$	$53.61_{\pm 2.12}$	$45.89_{\pm 1.24}$	**37.26**$_{\pm 0.82}$	$86.44_{\pm 2.18}$	$52.07_{\pm 0.89}$	$42.57_{\pm 1.18}$	$6.71_{\pm 0.55}$	$11.59_{\pm 0.82}$
ICIS	$63.43_{\pm 1.22}$	$21.65_{\pm 0.11}$	$52.14_{\pm 1.37}$	$52.21_{\pm 1.60}$	$70.07_{\pm 1.63}$	$59.84_{\pm 1.21}$	$0.01_{\pm 0.00}$	$96.24_{\pm 1.93}$	$0.02_{\pm 0.00}$	$42.11_{\pm 0.74}$	$28.56_{\pm 1.16}$	$34.04_{\pm 0.86}$
LaBo*	$50.83_{\pm 1.30}$	$43.99_{\pm 1.14}$	$48.22_{\pm 1.32}$	$46.89_{\pm 2.64}$	$57.45_{\pm 2.23}$	$51.64_{\pm 1.84}$	$33.07_{\pm 0.86}$	$80.40_{\pm 2.29}$	$46.86_{\pm 0.95}$	$43.81_{\pm 1.70}$	$7.52_{\pm 0.88}$	$12.84_{\pm 1.28}$
ZSLViT*	$64.17_{\pm 2.30}$	$52.04_{\pm 1.99}$	$50.63_{\pm 2.03}$	$53.44_{\pm 1.90}$	$72.21_{\pm 2.78}$	$61.42_{\pm 1.61}$	$30.39_{\pm 1.37}$	$82.95_{\pm 2.01}$	$44.48_{\pm 1.50}$	$41.11_{\pm 1.93}$	$26.59_{\pm 0.86}$	$32.29_{\pm 0.87}$
CoMC*	$56.29_{\pm 1.82}$	$49.28_{\pm 1.00}$	$44.27_{\pm 1.75}$	$10.21_{\pm 0.46}$	$77.81_{\pm 3.23}$	$18.05_{\pm 0.72}$	$3.45_{\pm 0.10}$	**96.29**$_{\pm 1.42}$	$6.66_{\pm 0.19}$	$36.40_{\pm 1.76}$	$25.51_{\pm 0.87}$	$30.00_{\pm 0.85}$
DFZSL*	$54.21_{\pm 1.12}$	$48.37_{\pm 1.04}$	$45.63_{\pm 1.26}$	$41.09_{\pm 0.93}$	$67.42_{\pm 1.77}$	$51.06_{\pm 0.88}$	$28.33_{\pm 0.72}$	$83.95_{\pm 2.01}$	$42.36_{\pm 0.84}$	$33.71_{\pm 0.66}$	$25.94_{\pm 0.92}$	$29.32_{\pm 0.64}$
FuDD*	$43.62_{\pm 1.32}$	$52.55_{\pm 1.97}$	$40.14_{\pm 0.44}$	$43.39_{\pm 1.20}$	$51.92_{\pm 1.14}$	$47.27_{\pm 0.85}$	$36.29_{\pm 0.85}$	$88.82_{\pm 1.73}$	$51.53_{\pm 0.90}$	$39.92_{\pm 0.92}$	$5.00_{\pm 0.00}$	$8.89_{\pm 0.02}$
GenZSL*	$45.50_{\pm 0.11}$	$51.95_{\pm 0.52}$	$51.44_{\pm 0.93}$	$45.50_{\pm 1.02}$	$51.95_{\pm 1.00}$	$48.51_{\pm 0.73}$	$35.77_{\pm 1.21}$	$84.07_{\pm 1.07}$	$50.19_{\pm 1.21}$	$41.62_{\pm 1.29}$	$37.18_{\pm 1.24}$	$39.27_{\pm 0.90}$
GEORWS	**71.28**$_{\pm 0.94}$	**60.34**$_{\pm 1.14}$	**59.07**$_{\pm 0.87}$	**65.05**$_{\pm 0.33}$	$80.08_{\pm 0.42}$	**71.79**$_{\pm 0.26}$	$37.12_{\pm 0.63}$	$96.21_{\pm 0.30}$	**53.57**$_{\pm 0.66}$	**48.61**$_{\pm 0.12}$	$45.33_{\pm 0.64}$	**46.91**$_{\pm 0.35}$

Proposition 1 separates two factors: how well the learned semantic coefficients select a point in the seen-weight convex hull, and how well this hull can approximate the ideal unseen classifier. This makes the main modeling assumption explicit.

The two added regularizers reduce coefficient instability during seen-class leave-one-out training. For a masked seen class m, if $\mathrm{KL}(p^W_{m,\cdot} \| \alpha_{m,\cdot}) \le \eta_m$, then Pinsker's inequality gives $\|p^W_{m,\cdot} - \alpha_{m,\cdot}\|_1 \le \sqrt{2\eta_m}$. Thus, $\mathcal{L}_{\mathrm{geo}}$ anchors the learned coefficients to a neighborhood distribution computed only from seen classifier weights. Similarly, if the per-class consistency loss is bounded by ξ_m, then $\|\hat{\mathbf{w}}_m - \tilde{\mathbf{w}}_m\|_2 \le \sqrt{\xi_m}$ and $\|\alpha_{m,\cdot} - \tilde{\alpha}_{m,\cdot}\|_1 \le \sqrt{2\xi_m}$. Thus, $\mathcal{L}_{\mathrm{cons}}$ limits changes in both reconstructed weights and affinity coefficients under semantic perturbations. Together, these results justify the objective: $\mathcal{L}_{\mathrm{align}}$ learns classifier synthesis, $\mathcal{L}_{\mathrm{geo}}$ ties coefficients to classifier geometry, and $\mathcal{L}_{\mathrm{cons}}$ reduces sensitivity to noisy pairwise descriptions.

4 Experiments

We evaluate I-ZSL on standard benchmark datasets that cover fine-grained, coarse-grained, and scene-level recognition: CUB [33], AWA2 [35], and SUN [24]. To ensure consistent evaluation, we adopt the ZSL and GZSL splits from [6]. We compare GEORWS with representative ZSL and I-ZSL baselines, including ConSE [22], COSTA [21], Sub.Reg.* [1], wDAE* [13], WAvg* [38], SMO* [38], ICIS [6], LaBo* [39], ZSLViT* [5], CoMC* [19], DFZSL* [32], FuDD* [10], and GenZSL* [3]. ICIS is the native I-ZSL comparison; ConSE and COSTA provide direct classifier-side comparisons. Methods marked with * are diagnostic weight-only adaptations, with unavailable image-dependent inputs or stages replaced or omitted. During expansion, no method uses images, stored or generated visual features, CLIP image embeddings, unseen validation data, or image-level accuracy for model selection. Dataset statistics, split details, and implementation details are provided in Appendix F.

4.1 Main Results

ZSL performance. Table 2 reports ZSL accuracy on CUB, AWA2, and SUN. GEORWS achieves the best ZSL performance on all three datasets. Compared with the strongest baselines, GEORWS improves ZSL accuracy by 7.11 percentage points on CUB, 4.67 percentage points on AWA2, and 6.93 percentage points on SUN. Compared with the native I-ZSL baseline ICIS [6], GEORWS also gives clear gains on fine-grained CUB and semantically overlapping SUN. These results suggest that pairwise semantic relations provide useful transfer signals beyond class-wise descriptions.

GZSL performance. Table 2 also reports unseen accuracy u, seen accuracy s, and harmonic mean H under GZSL. GEORWS achieves the strongest H on all three datasets. On CUB and SUN, the gains are large: GEORWS improves the best non-GEORWS harmonic mean by 10.37 and 7.64 percentage points, respectively. On AWA2, the margin is smaller: SMO* obtains a slightly higher unseen accuracy, while GEORWS achieves the best mean harmonic mean by maintaining stronger seen-class performance. No formal significance test was conducted. This result is important in GZSL,

Table 3: Ablation study of GEORWS. We report ZSL accuracy and GZSL unseen accuracy (u), seen accuracy (s), and harmonic mean (H).

Settings	Zero-Shot Accuracy			Generalized Zero-Shot Accuracy								
	CUB	AWA2	SUN	CUB			AWA2			SUN		
	Acc	Acc	Acc	u	s	H	u	s	H	u	s	H
GEORWS	**71.28**	**60.34**	**59.07**	**65.05**	**80.08**	**71.79**	37.12	96.21	**53.57**	**48.61**	**45.33**	**46.91**
w/o $\mathcal{L}_{\text{geo}}$	70.55	59.70	58.42	64.20	79.90	71.19	36.08	**96.30**	52.49	47.73	44.98	46.31
w/o $\mathcal{L}_{\text{cons}}$	70.71	59.83	58.55	64.55	79.95	71.43	36.60	96.22	53.03	48.02	45.01	46.47
w/o $\mathcal{L}_{\text{geo}}$ and $\mathcal{L}_{\text{cons}}$	70.39	59.92	58.28	64.29	79.93	71.26	36.40	96.29	52.83	47.76	44.77	46.22
w/o Affinity Description	63.85	53.42	51.27	55.12	78.31	64.70	31.22	94.78	46.97	40.83	42.95	41.86
w/ Class Description	64.23	55.23	57.83	59.00	78.72	67.45	33.23	95.27	49.27	45.52	43.05	44.25
w/o Semantic Encoding	61.74	52.63	49.81	50.48	77.91	61.26	28.37	92.64	43.44	37.55	41.33	39.35
$\lambda = 0.5$	67.89	57.71	55.36	60.41	79.26	68.56	34.61	95.11	50.75	45.92	43.68	44.77
$\lambda = 2$	66.22	56.38	54.19	58.74	78.32	67.13	33.84	96.12	50.06	44.15	41.47	42.77
Only MLLM (GPT-4)	11.37	47.82	26.94	8.42	62.15	14.83	**41.26**	69.37	51.74	18.73	47.62	26.89

Table 4: Robustness to semantic perturbations under GZSL. We report GZSL harmonic mean H.

Perturbation	GEORWS			w/o $\mathcal{L}_{\text{cons}}$		
	CUB	AWA2	SUN	CUB	AWA2	SUN
None	71.79	53.57	46.91	71.43	53.03	46.47
Embedding dropout	70.94	52.95	46.10	69.82	51.80	44.93
Gaussian noise	70.62	52.73	45.98	69.35	51.22	44.50
Text truncation	69.71	52.10	45.32	67.82	50.42	43.15
20% pair swap	68.35	51.38	44.40	65.70	49.85	41.80

where many baselines either collapse toward seen classes or improve unseen accuracy at the cost of much lower seen accuracy.

Fine-grained and scene-level recognition. The strongest gains are observed on CUB, where many classes are visually similar bird species. This supports the motivation that class-pair affinity descriptions expose fine-grained distinctions, while adaptive semantic encoding and classifier-geometry regularization help translate these relations into classifier-space coefficients. A similar pattern appears on SUN, where scene categories often have overlapping objects, context, and spatial layouts. Overall, the main results support three findings. First, pairwise descriptions improve transfer in fine-grained and semantically overlapping datasets. Second, adaptive semantic encoding is needed to align language-derived relations with classifier weights. Third, geometry and consistency regularization improve the stability of relational weight synthesis under the image-free constraint.

4.2 Further Analysis

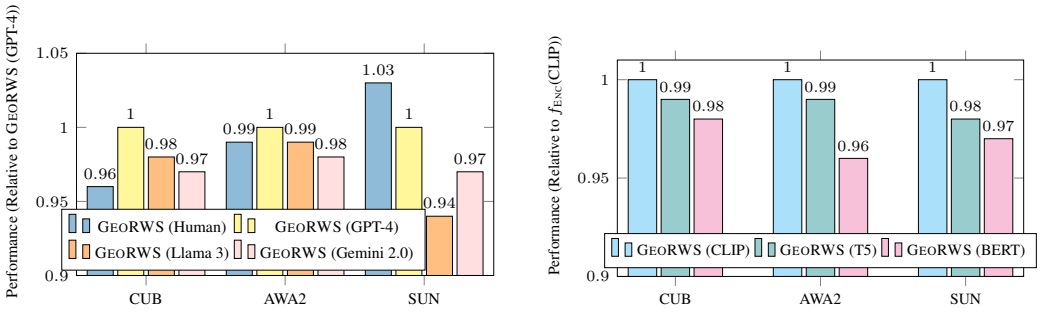

Figure 3: Relative performance of GEORWS with (a) different LLMs and human annotators, normalized to the GPT-4 variant within each dataset, and (b) different text encoders, normalized to the CLIP variant. A value of 1 denotes the corresponding reference performance.

Effect of regularizers. Table 3 evaluates the geometry and semantic-consistency regularizers. Removing $\mathcal{L}_{\text{geo}}$ reduces performance across datasets, indicating that classifier-space neighborhood information helps anchor the learned affinity coefficients. Removing $\mathcal{L}_{\text{cons}}$ also reduces performance,



Figure 4: Learned affinity matrices on CUB, AWA2, and SUN. Rows are unseen classes, columns are seen classes, and colors encode the synthesis coefficient $\alpha_{u,s}$; each row sums to one. In the AWA2 example, horse has its largest individual coefficients on zebra, ox, and deer. These are learned relations conditioned on classifier weights.

especially under GZSL, suggesting that consistency under semantic perturbation helps stabilize synthesized classifiers. The performance gains from these two regularizers are moderate, while the robustness gains under perturbed descriptions are larger. This suggests that the main role of the two regularizers is to stabilize coefficient estimation rather than replace the pairwise semantic signal.

Effect of removing class-pair affinity descriptions. Table 3 evaluates two variants that remove class-pair affinity descriptions by replacing them with label-only inputs or class-wise descriptions. The class-wise control uses descriptions from the same LLM and changes only the semantic input, keeping the remaining architecture and optimization settings fixed. Both variants degrade ZSL accuracy and GZSL harmonic mean. Class-wise descriptions are stronger than labels, but they remain weaker than pairwise affinity descriptions. This shows that explicit class-pair relations provide transfer cues that are not captured by independent class-level semantics.

Effect of removing adaptive semantic encoding. Removing the adaptive semantic encoder causes substantial performance drops on all datasets. This suggests that static text embeddings do not reliably align pairwise descriptions with the classifier space. The degradation is strongest on CUB and SUN, where fine-grained or context-overlapping categories require more task-specific semantic alignment.

Impact of λ. Table 3 studies the semantic reconstruction weight λ. The default value $\lambda = 1$ gives the most reliable trade-off across datasets. A smaller value weakens semantic preservation, while a larger value makes the encoder less flexible for classifier-space alignment. These results support a balanced objective that preserves semantic information while still adapting it to the classifier head.

LLMs as standalone classifiers. The direct multimodal LLM baseline is included as a diagnostic reference. It does not synthesize classifier weights and does not follow the read-only classifier expansion protocol. This baseline is highly dataset-dependent: it performs reasonably on coarse-grained AWA2 but poorly on fine-grained CUB and scene-level SUN. In contrast, GEORWS gives stable performance across all datasets because it uses LLM-generated descriptions as semantic side information rather than as final predictions.

Semantic sources and text encoders. Figure 3 evaluates the effect of different description sources and text encoders. The results show that GEORWS is not tied to a single LLM or a single frozen text encoder. High-quality LLM-generated descriptions generally perform well, while human-written descriptions remain competitive. Across text encoders, CLIP gives the strongest overall performance, with T5 close behind and BERT slightly weaker. These results suggest that stronger text representations help, but the adaptive semantic-to-classifier alignment remains important across different semantic sources.

Robustness to semantic perturbations. Table 4 evaluates the stability of GEORWS under perturbations of pairwise semantic descriptions. All perturbations are applied at inference to the pairwise semantic inputs, while model parameters are fixed. For a 20% pair swap, each description for (u, s) is independently replaced with probability 0.2 by that for (u', s), where u' is sampled uniformly from $\mathcal{U} \setminus \{u\}$; the reference s remains fixed. The full model degrades more slowly than the variant without $\mathcal{L}_{\text{cons}}$, especially under stronger perturbations such as text truncation and pair swaps. This supports

the role of semantic-consistency regularization in reducing sensitivity to noisy or unstable generated descriptions. Appendix G.7 reports the four-way regularizer comparison and additional corruption rates.

Transferability of generated class-wise descriptions. Appendix G.9 evaluates whether replacing the original class-wise descriptions of baselines with LLM-generated class-wise descriptions is sufficient to close the gap. This comparison is not input-matched to GEORWS, because GEORWS uses class-pair descriptions rather than class-wise descriptions. Instead, it tests whether class-wise text alone explains the observed gains. Most baselines improve only slightly or remain unstable, suggesting that the gains of GEORWS do not come only from text, but also from learning classifier-space affinity coefficients through leave-one-out reconstruction and geometry/consistency regularization.

Visualization of learned class-label relations. To better understand how GEORWS transfers information from seen to unseen classes, we visualize the learned affinity matrix $\alpha_{u,s}$ between unseen class labels and seen class labels. Figure 4 shows the complete learned relation matrices for CUB, AWA2, and SUN under the official I-ZSL splits. Each row corresponds to an unseen class, each column corresponds to a seen class, and the entry $\alpha_{u,s}$ indicates how strongly unseen class u draws on seen class s when synthesizing its classifier weight. Across all three datasets, the learned matrices exhibit clear semantic structure rather than diffuse or uniform weighting. On CUB, high-affinity regions are concentrated around semantically related bird categories, indicating that GEORWS learns localized transfer among visually and taxonomically similar classes. On AWA2, the matrix shows compact blocks that align with broad animal groups, such as aquatic mammals, terrestrial mammals, and small mammals. On SUN, although the label space is substantially larger, the learned affinities still show structured concentration patterns, suggesting that GEORWS captures meaningful relations among scene categories despite their greater diversity. Overall, these visualizations provide qualitative evidence that GEORWS synthesizes unseen classifiers by assigning larger weights to semantically related seen classes, consistent with the goal of relational weight synthesis.

5 Concluding Remarks

We study I-ZSL as a read-only classifier expansion problem and propose GEORWS, which uses pairwise semantics, classifier-space alignment, and geometry/consistency regularization to synthesize unseen-class classifier weights without using training images or image features during adaptation. Experiments on CUB, AWA2, and SUN show consistent gains over image-free and adapted zero-shot baselines, especially on fine-grained and semantically overlapping categories. Additional limitations, broader impact, and future directions are discussed in Appendix H.

Acknowledgments

This work is supported by funding from the Institute for Advanced Studies of the University of Luxembourg and the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Actions(GA #101081455). Additionally, we thank MeluXina for providing access to the supercomputer hosted by LuxProvide in Luxembourg, which was essential for conducting our computational experiments.

References

[1] Afra Feyza Akyürek, Ekin Akyürek, Derry Wijaya, and Jacob Andreas. Subspace regularizers for few-shot class incremental learning. In *Proceedings of the 2022 International Conference on Learning Representations (ICLR)*, 2022.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 2020 Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 1877–1901, 2020.

[3] Shiming Chen, Dingjie Fu, Salman Khan, and Fahad Shahbaz Khan. Genzsl: Generative zero-shot learning via inductive variational autoencoder. In *Proceedings of the 2025 International Conference on Machine Learning (ICML)*. PMLR, 2025.

[4] 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. In *Proceedings of the 2023 International Conference on Machine Learning (ICML)*, pages 4611–4622. PMLR, 2023.

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

[6] Anders Christensen, Massimiliano Mancini, A. Sophia Koepke, Ole Winther, and Zeynep Akata. Image-free classifier injection for zero-shot classification. In *Proceedings of the 2023 IEEE International Conference on Computer Vision (ICCV)*, pages 19026–19035, 2023.

[7] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2 edition, 2006.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 2009 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186. ACL, 2019.

[10] Reza Esfandiarpoor and Stephen H Bach. Follow-up differential descriptions: Language models resolve ambiguities for image classification. In *Proceedings of the 2024 International Conference on Learning Representations (ICLR)*, 2024.

[11] Ali Farhadi, Ian Endres, Derek Hoiem, and David A. Forsyth. Describing objects by their attributes. In *Proceedings of the 2009 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1778–1785. IEEE, 2009.

[12] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32(14-15):2627–2636, 1998.

[13] Spyros Gidaris and Nikos Komodakis. Generating classification weights with GNN denoising autoencoders for few-shot learning. In *Proceedings of the 2019 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21–30. IEEE, 2019.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the 2016 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

[15] 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 *Proceedings of the 2024 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23627–23637. IEEE, 2024.

[16] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(3):453–465, 2014.

[17] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[18] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Proceedings of the 2010 International Symposium on Circuits and Systems (ISCAS)*, pages 253–256. IEEE, 2010.

[19] Yicheng Liu, Jie Wen, Chengliang Liu, Xiaozhao Fang, Zuoyong Li, Yong Xu, and Zheng Zhang. Language-driven cross-modal classifier for zero-shot multi-label image recognition. In *Proceedings of the 2024 International Conference on Machine Learning (ICML)*, volume 235, pages 32173–32183. PMLR, 2024.

[20] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *Proceedings of the 2023 International Conference on Learning Representations (ICLR)*, 2023.

[21] Thomas Mensink, Efstratios Gavves, and Cees G. M. Snoek. COSTA: co-occurrence statistics for zero-shot classification. In *Proceedings of the 2014 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2441–2448. IEEE, 2014.

[22] Mohammad Norouzi, Tomás Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. In *Proceedings of the 2014 International Conference on Learning Representations (ICLR)*, 2014.

[23] Nicolas Papernot, Patrick D. McDaniel, Arunesh Sinha, and Michael P. Wellman. Sok: Security and privacy in machine learning. In *Proceedings of the 2018 IEEE European Symposium on Security and Privacy (Euro S&P)*, pages 399–414. IEEE, 2018.

[24] Genevieve Patterson, Chen Xu, Hang Su, and James Hays. The SUN attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision (IJCV)*, 108(1-2):59–81, 2014.

[25] Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, Xi-Zhao Wang, and Q. M. Jonathan Wu. A review of generalized zero-shot learning methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(4):4051–4070, 2023.

[26] Hongyu Qu, Jianan Wei, Xiangbo Shu, and Wenguan Wang. Learning clustering-based prototypes for compositional zero-shot learning. In *Proceedings of the 2025 International Conference on Learning Representations (ICLR)*, 2025.

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 2021 International Conference on Machine Learning (ICML)*, volume 139, pages 8748–8763. PMLR, 2021.

[28] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67, 2020.

[29] Bernardino Romera-Paredes and Philip H. S. Torr. An embarrassingly simple approach to zero-shot learning. In *Proceedings of the 2015 International Conference on Machine Learning (ICML)*, volume 37, pages 2152–2161. PMLR, 2015.

[30] Edgar Schönfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero- and few-shot learning via aligned variational autoencoders. In *Proceedings of the 2019 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8247–8255. IEEE, 2019.

[31] Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. Zero-shot learning through cross-modal transfer. In *Proceedings of the 2013 Annual Conference on Neural Information Processing Systems (NIPS)*, pages 935–943, 2013.

[32] Bowen Tang, Jing Zhang, Long Yan, Qian Yu, Lu Sheng, and Dong Xu. Data-free generalized zero-shot learning. In *Proceedings of the 2024 AAAI Conference on Artificial Intelligence (AAAI)*, pages 5108–5117, 2024.

[33] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *California Institute of Technology*, 2011.

[34] Wei Wang, Vincent W. Zheng, Han Yu, and Chunyan Miao. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):13:1–13:37, 2019.

[35] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 41(9):2251–2265, 2019.

[36] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proceedings of the 2018 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5542–5551. IEEE, 2018.

[37] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning - the good, the bad and the ugly. In *Proceedings of the 2017 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3077–3086. IEEE, 2017.

[38] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele, and Zeynep Akata. VGSE: visually-grounded semantic embeddings for zero-shot learning. In *Proceedings of the 2022 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9316–9325. IEEE, 2022.

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

A Prompt

A.1 Class-pair Affinity Description Generation Prompt

Example class-pair affinity description generation prompt for CUB/AWA2

Instruction: You are tasked with comparing two animal species based on their semantic relationships. Use your knowledge to generate a detailed and accurate description of the similarities and differences between the given animals. Ensure the comparison is domain-relevant and includes key characteristics that would help identify or distinguish these species in a real-world classification task.

Message: Describe the similarities and differences between [ANIMAL-1] and [ANIMAL-2]. Focus on key aspects such as their physical characteristics, habitat, diet, behavior, and any notable distinctions in their roles in the ecosystem. Provide a concise but detailed explanation that highlights how these animals are alike and how they differ.
Response: [The response will contain a structured and precise description of the relationship between the two animals, highlighting shared features and distinct attributes relevant to their classification.]

A.2 Class-wise Description Generation Prompt

Example class-wise description generation prompt for CUB/AWA2

Instruction: You are tasked with describing one animal species based on its characteristics. Use your knowledge to generate a detailed and accurate description of the given animal. Ensure the description is domain-relevant and includes key characteristics that would help identify this species in a real-world classification task.

Message: Describe the [ANIMAL]. Focus on key aspects such as its physical characteristics, habitat, diet, behavior, and any notable distinctions in its roles in the ecosystem.
Response: [The response will contain a structured and precise description of the given animal.]

A.3 Prompt for Using LLMs as Classifiers

Prompt for using LLMs as classifiers

Instruction: You are tasked with classifying an image into one of the given candidate categories. You are provided with the input image and a candidate set of class labels. Select the most appropriate class from this candidate set based on the visual content of the image. Provide only the final predicted class label without additional explanation.

Message: Input image: [IMAGE]
Candidate class labels: [CLASS-1, CLASS-2, ..., CLASS-N]
Which class does the image most likely belong to? Provide only the class label as the final answer.

Response: Predicted class: [CLASS-X]

B Example Generated Descriptions

This section presents the example generated descriptions from LLMs (GPT-4, Llama 3, and Gemini 2.0) and human annotators. Regarding human annotators, we ask 3 bachelor students with a computer science or mathematics background to perform this task.

CUB. *Coot vs. Grebe*

- *Human:* Coots and Grebes are both aquatic birds found in lakes and ponds, and both have lobed toes for swimming. Coots are dark-colored with a white beak and often feed on plants and small animals at the water's edge. Grebes have sleeker bodies, pointed bills, and are expert divers that mainly eat fish and insects. While both occupy similar habitats, coots are more likely to be seen walking on land, whereas grebes spend more time diving and rarely leave the water.
- *GPT-4:* Coot and Grebe are both freshwater diving birds with lobed feet and similar diets. Coots have a stocky body, blunt beak, and walk well on land. Grebes are sleek, have sharp beaks, and are clumsy on land. Coots are social; Grebes are more solitary and perform courtship displays.
- *Llama 3:* The Coot and the Grebe are two distinct waterbird species. Similarities include their aquatic lifestyle, freshwater habitats, and diet of aquatic plants and invertebrates.

  Differences lie in their physical characteristics, with Coots having a robust body, black plumage, and a white bill, whereas Grebes have a sleeker body and grayish or brownish plumage. Grebes are excellent divers, preferring open water, while Coots are more surface-oriented, favoring vegetated areas.

  Their behaviors also differ, with Grebes being more solitary during breeding and Coots forming larger groups. In the ecosystem, both species play important roles, but Coots have a greater impact on vegetation due to their plant-based diet. Overall, while sharing some similarities, the Coot and the Grebe exhibit distinct adaptations to their environments.
- *Gemini 2.0:* Similarities: Aquatic birds, swim/dive, eat aquatic invertebrates/some plants, build floating nests.

  Differences: Coots: Dark plumage, white bill & frontal shield, lobed toes (can walk on land), omnivorous, often in flocks. Grebes: Streamlined, pointed bill, distinctly lobed feet (poor walking), mainly carnivorous (fish/invertebrates), often solitary/pairs. Key Distinctions: Bill/foot morphology and terrestrial ability.

CUB. *Wren vs. Finch*

- *Human:* Wrens and Finches are both small songbirds commonly found in gardens and forests. Wrens have thin, slightly curved bills, short wings, and are known for their upright tails and insect-based diet. Finches have stout, conical bills adapted for eating seeds, and generally display a more robust body shape. While both may sing and nest in shrubs, wrens are more active in searching for insects, while finches mainly forage for seeds on the ground or in trees.
- *GPT-4:* Wren and Finch are both small songbirds often found in gardens and woodlands. Wrens are brown, with short wings and tails held upright; they eat insects and are very active. Finches are more colorful, with conical beaks for seeds, and show more varied plumage. Wrens are solitary; Finches often form flocks.
- *Llama 3:* The Wren and the Finch are two distinct bird species. Similarities include their small size, insectivorous and seed-eating diets, and social behaviors, often forming flocks.

  Differences lie in their physical characteristics, with Wrens having a rounder body, shorter tail, and longer bill, whereas Finches have a more elongated body, notched tail, and shorter bill. Wrens are also known for their bold and loud songs, whereas Finches have more melodic and varied songs.

  In terms of habitat, Wrens prefer denser vegetation, such as shrubs and trees, while Finches are found in a wider range of habitats, including backyards, fields, and woodlands. Overall, while sharing some similarities, the Wren and the Finch exhibit distinct characteristics and preferences.

- *Gemini 2.0:* Similarities: Small passerine birds, diurnal, known for songs.

  Differences: Wrens: Tiny, roundish, thin bill, short tail (often upright), mainly brown, primarily insectivorous, often solitary, dome-shaped nests. Finches: Slightly larger, stout, conical bill, varied tail, often colorful, primarily granivorous, sometimes in flocks, cup-shaped nests. Key Distinctions: Bill shape (thin vs. conical) and primary diet (insects vs. seeds).

AWA2. *Wolf vs. Domestic Dog*

- *Human:* Wolves and domestic dogs are closely related canines, both with similar body structures and social behaviors. Wolves are wild animals with larger bodies, longer legs, and thick, grayish fur; they live and hunt in packs and rely on hunting large prey in forests or open lands. Domestic dogs are smaller on average, have diverse fur colors and shapes due to selective breeding, and live with humans, eating a variety of foods. Unlike wolves, dogs rely on humans for care and rarely hunt in packs.
- *GPT-4:* Wolf and Domestic Dog share ancestry and body plan. Wolves are larger, have broader skulls, and stronger jaws. They live in wild packs and hunt large prey. Dogs vary in size and shape due to breeding, live with humans, and depend on them for food. Wolves avoid humans; dogs are domesticated and socialized.
- *Llama 3:* The Wolf and the Domestic Dog are two closely related species. Similarities include their carnivorous diet, pack behavior, and loyal nature.

  Differences lie in their physical characteristics, with Wolves having a more muscular build, sharper teeth, and a thicker coat, whereas Domestic Dogs have a more varied size, shape, and coat type due to breeding. Wolves are also more territorial and have a stronger prey drive, whereas Domestic Dogs are more socialized to humans and have a wider range of behaviors and roles, from companions to working animals. Overall, while sharing a common ancestor, Wolves and Domestic Dogs have distinct differences in their physiology, behavior, and relationship with humans.
- *Gemini 2.0:* Similarities: Same species (Canis lupus), pack-oriented, quadrupedal carnivores, similar vocalizations.

  Differences: Wolves: Larger, powerful build, straighter tail, muted colors, more cautious/independent, shorter socialization. Domestic Dogs: Wide size/shape range, varied tail, diverse colors, more amenable to humans, longer socialization. Key Distinctions: Physical variability/size, temperament/dependence on humans.

AWA2. *Horse vs. Zebra*

- *Human:* Horses and zebras are both hoofed mammals in the Equidae family. Horses have solid-colored coats, are domesticated, and are used for riding, transport, or farm work. Zebras are wild animals, easily identified by their black-and-white stripes, and live in herds on African grasslands. While both mainly eat grasses, horses are more adaptable to different environments, while zebras rely on their stripes for camouflage and social behavior in the wild.
- *GPT-4:* Horse and Zebra are both equids with similar body shape and herbivorous diet. Horses are domesticated, come in various colors, and are used for riding and work. Zebras are wild, have black-and-white stripes, and are more aggressive and harder to tame. Horses neigh and are social with humans; zebras bark and live in tight herds for predator defense.
- *Llama 3:* The Horse and the Zebra are two closely related species. Similarities include their herbivorous diet, social behavior, and grazing habits.

  Differences lie in their physical characteristics, with Horses having a solid coat color and Zebras having distinctive black and white stripes. Zebras are also more adapted to wilderness habitats and have a stronger instinct for self-preservation, whereas Horses are more domesticated and have a wider range of behaviors and roles, from companions to working animals. Overall, while sharing a common ancestor, Horses and Zebras have distinct differences in their physiology, behavior, and relationship with their environment.
- *Gemini 2.0:* Similarities: Equids, herbivorous, similar body plan (hooves, legs, tail, mane), live in herds.

Differences: Zebras: Black and white stripes, stockier build, coarser mane, less domesticable, native to Africa. Horses: Typically solid-colored coats (or spots), more varied build, often longer mane, generally domesticable, global distribution (wild populations more restricted). Key Distinction: Coat pattern (stripes vs. solid/spots).

SUN. *Classroom vs. Library*

- *Human:* Classrooms and libraries are both indoor environments designed for learning. Classrooms typically contain desks, a teacher's area, and boards for instruction, and are used for active teaching and group discussions. Libraries are quieter spaces filled with bookshelves, reading tables, and study areas, intended for reading, research, and independent study. While both support education, classrooms focus on direct instruction, while libraries provide resources and a quiet atmosphere for individual learning.
- *GPT-4:* Classroom and Library are both indoor educational spaces. A classroom is for instruction, with a teacher, desks, and active discussion. A library is for independent study, with books, quiet zones, and minimal interaction. Classrooms focus on teaching; libraries support learning through resources.
- **Llama 3:** Classrooms and Libraries are two indoor educational spaces that both support learning, reading, and information access. Similarities include the presence of desks or tables, seating areas, quiet study activities, and frequent use by students and educators.

  Differences lie in their layout and primary function. Classrooms are usually organized for structured instruction, with a teacher's area, whiteboard or projector, rows of desks, and scheduled group activities. Libraries are organized around bookshelves, reading tables, catalog or search areas, and spaces for independent study or research. Classrooms are typically more interactive and teacher-led, whereas libraries emphasize self-directed learning, quiet reading, and access to learning resources. Overall, while both belong to educational environments, they can be distinguished by structured instruction in classrooms versus resource-centered study in libraries.
- *Gemini 2.0:* Similarities: Spaces for learning, contain informational resources, found in educational settings.

  Differences: Classroom: For structured teaching with an instructor, curriculum-focused, scheduled attendance. Library: For self-directed learning, diverse resources, voluntary access. Key Distinction: Presence of a teacher/structured instruction vs. self-directed resource access.

SUN. *Beach vs. Desert*

- *Human:* Beaches and deserts are both natural environments with sandy ground. Beaches are located next to bodies of water like oceans or lakes, have moist sand, support aquatic life, and are often visited for recreation. Deserts are dry, receive very little rainfall, have arid sand or rocky soil, and support sparse, drought-resistant vegetation and animals. While both can be hot and sunny, beaches are shaped by water and tides, while deserts are shaped by wind and extreme dryness.
- *GPT-4:* Beach and Desert are both sandy environments but differ in key aspects. Beaches are coastal, humid, and support marine life; deserts are inland, arid, and support drought-adapted species. Beaches have tides and moderate temperatures; deserts have extreme temperature swings and minimal water.
- *Llama 3:* The Beach and the Desert are two distinct ecosystems. Similarities include their ability to support unique and adapted plant and animal species.

  Differences lie in their climate, geography, and characteristics, with Beaches being coastal areas with sandy or rocky shores, salty air, and a variety of marine life, whereas Deserts are arid, landlocked regions with extreme temperatures, limited vegetation, and specialized wildlife. Beaches are also typically characterized by high humidity and abundant water, whereas Deserts are defined by low humidity and scarce water resources. Overall, the Beach and the Desert exhibit contrasting environments and support different types of biodiversity.
- *Gemini 2.0:* Similarities: Sparse vegetation, large areas of sediment, can have extreme temperatures, subject to wind/water erosion.

Differences: Beach: Adjacent to water, higher water availability/humidity, vegetation adapted to salt/waves, temperature moderated by water, shaped by water. Desert: Arid (low water), vegetation adapted to drought, extreme daily temperature swings, shaped by wind. Key Distinction: Presence/absence of significant water.

Costs of LLM Usages. In our pipeline, we issue one LLM request per *ordered target–reference pair* to generate one class-pair affinity description. For a dataset with $|S|$ seen classes and $|U|$ unseen classes, this results in $|S|(|S|-1)$ seen–seen pairs for the leave-one-out training and $|U||S|$ unseen–seen pairs for inference. Using the standard splits (CUB: $|S|$=150, $|U|$=50; AWA2: $|S|$=40, $|U|$=10; SUN: $|S|$=645, $|U|$=72), the total number of queried pairs is $150 \cdot 149 + 50 \cdot 150 = 29{,}850$ (CUB), $40 \cdot 39 + 10 \cdot 40 = 1{,}960$ (AWA2), and $645 \cdot 644 + 72 \cdot 645 = 461{,}820$ (SUN), for a total of $493{,}630$ requests. We use GPT-4 for the main description generation, and Gemini 2.0 for the LLM ablation, resulting in $493{,}630$ queries to GPT-4 and $493{,}630$ queries to Gemini 2.0. Each request uses a prompt of approximately 150 tokens and receives a model response of about 100 tokens, for a total of roughly 250 tokens per request. Under the pricing assumptions used for these estimates (\$0.4 per 1M prompt tokens and \$1.6 per 1M completion tokens for GPT-4; \$0.3 per 1M prompt/completion tokens for Gemini 2.0), the average cost per GPT-4 request is $150 \times 0.4/10^6 + 100 \times 1.6/10^6 \approx \$2.2 \times 10^{-4}$, and per Gemini 2.0 request is $250 \times 0.3/10^6 \approx \$7.5 \times 10^{-5}$. Summing across all calls, the total estimated cost is about $\$108.60$ for GPT-4 and $\$37.02$ for Gemini 2.0, for a combined API cost of less than $\$146$. These are token-based estimates, not billed amounts. Most costs are attributable to GPT-4, which performs the main description generation, while Gemini 2.0 is used for ablation studies. Local experiments with Llama 3 incur no additional API charges. These results show that, for our experimental scale and prompt length, LLM-based description generation remains practically affordable. We publish our generated descriptions and generated text embeddings in our project at `https://github.com/zhiqiangzhongddu/GeoRWS`.

Descriptions and embeddings are generated once and cached for a fixed label set; no repeated LLM calls are needed during optimization, subsequent weight synthesis, or per-image prediction. Both seen–seen directions are stored because the scorer distinguishes target and reference roles. Adding one unseen class requires $|S|$ new descriptions. Across the three datasets, storing $493{,}630$ embeddings of dimension 512 in float32 requires $1{,}010{,}954{,}240$ bytes (about 1.01 GB), excluding text and serialization metadata. Quadratic seen–seen preprocessing remains a limitation; reusing symmetric descriptions is a possible, unevaluated reduction.

C Algorithm

Algorithm 1 summarizes the training and inference procedure of GEORWS. During training, each seen class is masked in turn and reconstructed from the remaining seen classifier weights using pairwise semantic descriptions. The objective combines classifier-weight alignment, semantic reconstruction, classifier-geometry regularization, and semantic-consistency regularization. During inference, the learned semantic-to-affinity mapping is applied to unseen classes to synthesize unseen classifier weights, which are then injected into the classifier head.

Phase 1: Training on seen classes. Training uses leave-one-out reconstruction over seen classes. For each masked seen class $m \in \mathcal{S}$ and each reference class $s \in \mathcal{S} \setminus \{m\}$, the frozen LLM generates a class-pair affinity description, and the frozen text encoder maps it to an embedding $\mathbf{a}_{m,s}$. The trainable semantic encoder maps this embedding to $\mathbf{z}_{m,s} = f_\theta(\mathbf{a}_{m,s})$. The scoring head then computes an affinity logit

$$g_{m,s} = f_\psi(\mathbf{z}_{m,s} \oplus \mathbf{w}_s),$$

where the concatenation with $\mathbf{w}_s$ conditions semantic affinity estimation on the classifier-space prototype of the reference class. A softmax over all remaining seen classes gives coefficients

$$\alpha_{m,s} = \frac{\exp(g_{m,s})}{\sum_{s' \in \mathcal{S} \setminus \{m\}} \exp(g_{m,s'})}, \qquad s \in \mathcal{S} \setminus \{m\}.$$

The masked classifier weight is reconstructed as

$$\hat{\mathbf{w}}_m = \sum_{s \in \mathcal{S} \setminus \{m\}} \alpha_{m,s} \mathbf{w}_s.$$

The alignment loss $\mathcal{L}_{\text{align}}$ matches $\hat{\mathbf{w}}_m$ to the true seen-class weight $\mathbf{w}_m$, while the reconstruction loss $\mathcal{L}_{\text{recon}}$ keeps the latent semantic embedding informative by decoding $\mathbf{z}_{m,s}$ back to $\mathbf{a}_{m,s}$.

The geometry loss $\mathcal{L}_{\text{geo}}$ regularizes the learned coefficients using the local geometry of the classifier head. For each masked class m, we form a teacher distribution $p^W_{m,\cdot}$ from cosine similarities between $\mathbf{w}_m$ and the remaining seen-class weights:

$$p^W_{m,s} = \frac{\exp(\cos(\mathbf{w}_m, \mathbf{w}_s)/\tau_W)}{\sum_{s' \in \mathcal{S} \setminus \{m\}} \exp(\cos(\mathbf{w}_m, \mathbf{w}_{s'})/\tau_W)}, \qquad s \in \mathcal{S} \setminus \{m\}.$$

Minimizing

$$\mathrm{KL}\big(p^W_{m,\cdot} \,\|\, \alpha_{m,\cdot}\big)$$

encourages the semantic coefficients to remain consistent with classifier-space neighborhoods.

The consistency loss $\mathcal{L}_{\text{cons}}$ improves robustness to noisy or unstable semantic descriptions. Given an augmented embedding $\tilde{\mathbf{a}}_{m,s} = \mathrm{Aug}(\mathbf{a}_{m,s})$, the model computes another coefficient vector $\tilde{\alpha}_{m,\cdot}$ and another reconstructed weight $\tilde{\mathbf{w}}_m$ using the same encoder, scoring head, and softmax normalization over $\mathcal{S} \setminus \{m\}$. The loss penalizes changes in both the reconstructed classifier weight and the coefficient distribution:

$$\mathcal{L}_{\text{cons}} = \sum_{m \in \mathcal{S}} \|\hat{\mathbf{w}}_m - \tilde{\mathbf{w}}_m\|_2^2 + \sum_{m \in \mathcal{S}} D_{\text{SKL}}(\alpha_{m,\cdot}, \tilde{\alpha}_{m,\cdot}),$$

where

$$D_{\text{SKL}}(p, q) = \mathrm{KL}(p \| q) + \mathrm{KL}(q \| p).$$

In the main experiments, $\mathrm{Aug}(\cdot)$ is implemented as embedding dropout applied to $\mathbf{a}_{m,s}$ during training. Description-level perturbations, such as text truncation and pair swapping, are used only for robustness analysis. The full objective is

$$\mathcal{L} = \mathcal{L}_{\text{align}} + \lambda \mathcal{L}_{\text{recon}} + \beta \mathcal{L}_{\text{geo}} + \gamma \mathcal{L}_{\text{cons}}.$$

Phase 2: Inference via classifier injection. After training, the learned semantic-to-affinity mapping is applied to unseen classes. For each unseen class $u \in \mathcal{U}$ and each seen class $s \in \mathcal{S}$, the method retrieves the cached class-pair affinity embedding, $\mathbf{a}_{u,s} = f_{\text{ENC}}\big(f_{\text{LLM}}(u, s)\big)$, and computes

$$g_{u,s} = f_\psi(f_\theta(\mathbf{a}_{u,s}) \oplus \mathbf{w}_s).$$

The logits are normalized over all seen classes:

$$\alpha_{u,s} = \frac{\exp(g_{u,s})}{\sum_{s' \in \mathcal{S}} \exp(g_{u,s'})}, \qquad s \in \mathcal{S}.$$

The unseen-class classifier weight is then synthesized as

$$\mathbf{w}_u = \sum_{s \in \mathcal{S}} \alpha_{u,s} \mathbf{w}_s.$$

All synthesized unseen weights are injected into the classifier head to form

$$\mathbf{W}_{\mathcal{Y}} = \mathbf{W}_{\mathcal{S}} \cup \mathbf{W}_{\mathcal{U}}.$$

Given a test image x, prediction uses cosine scores with the fixed feature extractor:

$$\hat{\Phi}(x) = \arg\max_{y \in \mathcal{Y}} N(\mathbf{w}_y)^\top N(\mathcal{F}(x)).$$

Here N is the zero-safe normalization defined in Section 3.5; the synthesized weights above remain unnormalized.

Training and inference example. On AWA2, the training procedure masks wolf and reconstructs its classifier from the remaining 39 seen classes. In the stored qualitative run, German shepherd, fox, and collie receive the three largest coefficients. At inference, the same learned mapper synthesizes the unseen horse classifier from all 40 seen classes; zebra, ox, and deer are its largest contributors. In both cases, the other reference classes retain their learned contributions. Training compares the reconstructed wolf weight with its known weight, whereas inference appends the synthesized horse weight without an unseen-class reconstruction target.

D Related Work Discussion

Table 1 provides a side-by-side comparison of GEORWS and existing approaches in I-ZSL. We highlight three key dimensions relevant to I-ZSL: the need for manually defined class descriptions, the use of adaptive semantic guidance, and the reliance on image vision features.

Most early methods, such as ConSE [22] and COSTA [21], support I-ZSL but require class-level semantic information and do not employ adaptive semantic modeling. Methods like wDAE [13] and WAvg [38] do not support I-ZSL at all, since they depend on visual features for both training and inference. ICIS [6] removes the dependency on images during adaptation and learns mappings between semantic representations and classifier weights; it can use attributes or encoded class names.

GEORWS combines the following properties:

- No reliance on human-written class descriptions: All semantic information is generated automatically by LLMs, removing the need for expert annotation and enabling adaptation to new domains where such information is unavailable or difficult to collect.
- Adaptive semantic guidance: Our model introduces a task-adaptive encoder trained using the fixed seen-class classifier weights. This design allows the semantic space to be optimized for the target task, improving discrimination between fine-grained classes.
- GEORWS does not require training images or extracted training-image features during adaptation. At test time, it uses the fixed deployed feature extractor with the extended classifier head.

Our approach combines image-free adaptation with automatically generated pairwise descriptions and adaptive semantic modeling. This enables a more flexible and robust framework for I-ZSL, as summarized in Table 1. Our results demonstrate that these design choices translate to clear improvements in both accuracy and generalization, particularly in fine-grained or specialized domains where existing assumptions are not realistic.

E Additional Theoretical Analysis

This appendix provides the proofs for the results stated in Section 3.6. We use the notation from the main text. For a set $\mathcal{C} \subseteq \mathbb{R}^D$, define

$$\operatorname{dist}(\mathbf{w}, \mathcal{C}) = \inf_{\mathbf{v} \in \mathcal{C}} \|\mathbf{w} - \mathbf{v}\|_2.$$

For two probability vectors p and q, define the symmetric KL divergence as

$$D_{\mathrm{SKL}}(p, q) = \mathrm{KL}(p\|q) + \mathrm{KL}(q\|p).$$

All KL divergences use natural logarithms. We use Pinsker's inequality [7], which states that for probability vectors p and q over the same finite support,

$$\|p - q\|_1 \leq \sqrt{2\,\mathrm{KL}(p\|q)}.$$

E.1 Proof of Proposition 1

Proof. The synthesized unseen classifier is

$$\mathbf{w}_u = \sum_{s \in \mathcal{S}} \alpha_{u,s} \mathbf{w}_s,$$

where $\alpha_{u,\cdot}$ is produced by a softmax over $\mathcal{S}$. Thus,

$$\alpha_{u,s} \geq 0, \quad \sum_{s \in \mathcal{S}} \alpha_{u,s} = 1.$$

Therefore,

$$\mathbf{w}_u \in \mathcal{C}_{\mathcal{S}} = \operatorname{conv}\left(\{\mathbf{w}_s : s \in \mathcal{S}\}\right).$$

By definition,

$$\mathbf{w}_u^{\circ} = \arg\min_{\mathbf{w} \in \mathcal{C}_{\mathcal{S}}} \|\mathbf{w} - \mathbf{w}_u^{\dagger}\|_2.$$

Since $\mathcal{C}_{\mathcal{S}}$ is the convex hull of finitely many vectors, it is compact, so the minimizer exists. Moreover,

$$\|\mathbf{w}_u^{\circ} - \mathbf{w}_u^{\dagger}\|_2 = \operatorname{dist}\left(\mathbf{w}_u^{\dagger}, \mathcal{C}_{\mathcal{S}}\right).$$

Using the triangle inequality,

$$\begin{aligned} \|\mathbf{w}_u - \mathbf{w}_u^{\dagger}\|_2 &= \|\mathbf{w}_u - \mathbf{w}_u^{\circ} + \mathbf{w}_u^{\circ} - \mathbf{w}_u^{\dagger}\|_2 && (4)\\ &\leq \|\mathbf{w}_u - \mathbf{w}_u^{\circ}\|_2 + \|\mathbf{w}_u^{\circ} - \mathbf{w}_u^{\dagger}\|_2 && (5)\\ &= \|\mathbf{w}_u - \mathbf{w}_u^{\circ}\|_2 + \operatorname{dist}\left(\mathbf{w}_u^{\dagger}, \mathcal{C}_{\mathcal{S}}\right). && (6) \end{aligned}$$

This proves Equation 3. □

E.2 Classifier-geometry Regularization

Lemma 1 (Coefficient control by geometry regularization)**.** *For a masked seen class m, suppose*

$$\mathrm{KL}\left(p_{m,\cdot}^{W} \,\|\, \alpha_{m,\cdot}\right) \leq \eta_m.$$

Then

$$\left\|p_{m,\cdot}^{W} - \alpha_{m,\cdot}\right\|_1 \leq \sqrt{2\eta_m}.$$

Proof. Both $p_{m,\cdot}^{W}$ and $\alpha_{m,\cdot}$ are probability distributions over the same support $\mathcal{S} \setminus \{m\}$. By Pinsker's inequality,

$$\left\|p_{m,\cdot}^{W} - \alpha_{m,\cdot}\right\|_1 \leq \sqrt{2\,\mathrm{KL}\left(p_{m,\cdot}^{W} \,\|\, \alpha_{m,\cdot}\right)}.$$

Using the assumed bound gives

$$\left\|p_{m,\cdot}^{W} - \alpha_{m,\cdot}\right\|_1 \leq \sqrt{2\eta_m}.$$

□

E.3 Semantic-consistency Regularization

Lemma 2 (Stability under semantic perturbation)**.** *For a masked seen class m, suppose the per-class consistency term satisfies*

$$\|\hat{\mathbf{w}}_m - \tilde{\mathbf{w}}_m\|_2^2 + D_{\mathrm{SKL}}\left(\alpha_{m,\cdot}, \tilde{\alpha}_{m,\cdot}\right) \leq \xi_m.$$

Then

$$\|\hat{\mathbf{w}}_m - \tilde{\mathbf{w}}_m\|_2 \leq \sqrt{\xi_m},$$

and

$$\|\alpha_{m,\cdot} - \tilde{\alpha}_{m,\cdot}\|_1 \leq \sqrt{2\xi_m}.$$

Proof. Since both terms in the consistency bound are nonnegative,

$$\|\hat{\mathbf{w}}_m - \tilde{\mathbf{w}}_m\|_2^2 \leq \xi_m.$$

Taking square roots gives

$$\|\hat{\mathbf{w}}_m - \tilde{\mathbf{w}}_m\|_2 \leq \sqrt{\xi_m}.$$

We also have

$$D_{\mathrm{SKL}}\left(\alpha_{m,\cdot}, \tilde{\alpha}_{m,\cdot}\right) \leq \xi_m.$$

By definition,

$$D_{\mathrm{SKL}}\left(\alpha_{m,\cdot}, \tilde{\alpha}_{m,\cdot}\right) = \mathrm{KL}\left(\alpha_{m,\cdot} \,\|\, \tilde{\alpha}_{m,\cdot}\right) + \mathrm{KL}\left(\tilde{\alpha}_{m,\cdot} \,\|\, \alpha_{m,\cdot}\right).$$

Both KL terms are nonnegative, so

$$\mathrm{KL}\left(\alpha_{m,\cdot} \,\|\, \tilde{\alpha}_{m,\cdot}\right) \leq \xi_m.$$

Applying Pinsker's inequality gives

$$\|\alpha_{m,\cdot} - \tilde{\alpha}_{m,\cdot}\|_1 \leq \sqrt{2\,\mathrm{KL}\left(\alpha_{m,\cdot} \,\|\, \tilde{\alpha}_{m,\cdot}\right)} \leq \sqrt{2\xi_m}.$$

□

Proposition 1 separates the synthesis error into a coefficient-estimation term and a convex-hull approximation term. Lemma 1 shows that the classifier-geometry loss ties the learned coefficients to local neighborhoods in the observed classifier head. Lemma 2 shows that the semantic-consistency loss gives stability control under perturbations of the pairwise semantic embeddings. Together, these results support the objective in Equation 2.

F Experimental Settings

F.1 Datasets

We evaluate I-ZSL performance on three widely used *Zero-Shot Learning* (ZSL) benchmark datasets: CUB, AWA2, and SUN. These datasets span fine-grained and coarse-grained classification tasks, enabling a comprehensive evaluation of GEORWS under different levels of semantic granularity. To ensure consistency with prior work, we adopt the class splits for seen and unseen categories as defined in [6] for both the ZSL and *Generalized Zero-Shot Learning* (GZSL) settings.

- **CUB-200-2011 (CUB)** [33]: A fine-grained bird classification dataset containing 200 categories of North American bird species. Each category is annotated with extensive semantic information, including part-level attributes and textual descriptions. Following established ZSL benchmarks [36, 37], we use 150 classes as seen and 50 as unseen. Since GEORWS operates in an image-free setting, we utilize class-pair affinity descriptions and classifier injections for the I-ZSL training process [6]. The high intra-class similarity in CUB makes it a challenging dataset, as accurate classification requires fine-grained semantic reasoning.
- **Animals with Attributes 2 (AWA2)** [35]: A large-scale coarse-grained animal classification dataset comprising 50 mammal species, each annotated with 85 attributes covering physical characteristics, habitat, and behavior. We adopt the standard split of 40 seen and 10 unseen classes. Unlike CUB, AWA2 exhibits greater inter-class variation, making it well-suited for evaluating the ability of GEORWS to generalize across distinct semantic concepts. The attribute-based structure of AWA2 is particularly relevant for I-ZSL, as semantic information must fully compensate for the absence of visual data.
- **SUN Attribute (SUN)** [24]: A large-scale scene classification dataset containing 717 categories covering diverse indoor and outdoor environments. Each scene category is annotated with 102 attributes, such as object presence, spatial arrangement, and lighting conditions. We follow the standard protocol in [16], using 645 seen and 72 unseen classes, consistent with the ZSL/GZSL split from [35]. SUN is particularly challenging for I-ZSL, as scene categories often lack distinct object-based semantics, requiring a deeper understanding of high-level contextual attributes.

These datasets provide a diverse evaluation setting for I-ZSL. **CUB** assesses the ability to capture fine-grained semantic variations among visually similar species, **AWA2** evaluates generalization across distinct animal classes with structured attribute descriptions, and **SUN** tests the ability to infer semantic relationships from abstract contextual attributes. By encompassing different levels of semantic complexity, our evaluation ensures a rigorous assessment of GEORWS's capacity to recognize unseen classes without relying on visual information.

F.2 Competing Methods

We compare GEORWS against multiple state-of-the-art methods for ZSL and I-ZSL. This comparative analysis highlights the key differences in semantic representation, classifier injection, and adaptation mechanisms across these approaches.

1. **ConSE** [22] forms a semantic embedding for a test image by weighting seen-class semantic vectors with the classifier's prediction probabilities. Instead of explicitly learning semantic embeddings, it aggregates seen-class predictions in a weighted manner, making it an effective but limited baseline for ZSL due to its reliance on the quality of seen-class classifiers.
2. **COSTA** [21] transfers classifiers using semantic relationships obtained from label co-occurrence statistics. It estimates an unseen classifier from known classifiers without requiring visual examples of the unseen class.
3. **Sub.Reg.*** [1] introduces a subspace regularization approach that constrains the embedding space through additional priors, enhancing the preservation of semantic structures across class distributions. By improving the alignment between seen and unseen class spaces, it increases generalization but still requires access to seen-class features.

4. **wDAE*** [13] employs a denoising autoencoder (DAE) architecture to reconstruct classifier weights. It uses an auxiliary decoding mechanism to refine these weights and improve their robustness. Despite its generative capabilities, wDAE requires visual data during training, restricting its applicability in image-free settings.

5. **WAvg*** [38] models unseen-class representations as weighted averages of seen-class embeddings. It assumes that unseen categories exist as linear combinations of known ones. While computationally efficient, this assumption is often too simplistic for capturing fine-grained distinctions in complex classification tasks.

6. **SMO*** [38] uses *Similarity Matrix Optimization* in VGSE to transfer semantic information from seen to unseen classes. The original method learns visually grounded semantic embeddings from seen images, so the weight-only variant is a diagnostic adaptation.

7. **ICIS** [6] is a native I-ZSL method that learns mappings between class semantics and classifier weights through reconstruction objectives. It can use class names or attributes and does not require images during classifier expansion.

8. **LaBo*** [39] constructs *Language Guided Bottlenecks* from LLM-generated concepts and learns a concept-based classifier using visual features. Our weight-only comparison omits unavailable image-dependent stages and does not reproduce the full original method.

9. **ZSLViT** [5] leverages a transformer-based vision model, *Vision Transformer (ViT)*, for ZSL. While it moves beyond traditional convolutional-based architectures, ZSLViT still requires access to visual data during adaptation, making it unsuitable for fully image-free scenarios.

10. **CoMC** [19] learns a language-driven cross-modal classifier for zero-shot multi-label image recognition. Its original prediction procedure uses vision-language image features; our classifier-side adaptation is a diagnostic comparison.

11. **DFZSL*** [32] addresses the *data-free zero-shot learning* problem, where access to raw training images is restricted but a base-class classifier is available. It reconstructs virtual features by modeling the class-wise distribution with a von Mises-Fisher (vMF) prior and aligns them with text features through a feature-language prompt tuning strategy. Our diagnostic adaptation retains classifier-side alignment and omits visual-feature generation and image-encoder fine-tuning.

12. **FuDD** [10] uses *Follow-Up Differential Descriptions* to distinguish classes identified as ambiguous for a test image. FuDD* omits this image-conditioned ambiguity-detection stage and retains class-level semantic information for the diagnostic comparison.

13. **GenZSL*** [3] proposes an inductive variational autoencoder for generative zero-shot learning. Instead of generating unseen visual features from scratch using expert-defined semantic vectors, GenZSL induces unseen-class samples from semantically similar seen classes using weak class semantic vectors derived from class names, such as CLIP text embeddings. It further improves generated samples through class diversity promotion and target class-guided information boosting. Although GenZSL is efficient and effective in standard generative ZSL, it still depends on visual feature synthesis and training a classifier on generated visual features. In our I-ZSL adaptation, we replace image-feature inputs with seen-class classifier weights where applicable, which makes the comparison compatible with the image-free protocol but limits the full generative mechanism of GenZSL.

Adapt image vision features models to I-ZSL settings. Among these methods, ICIS is the most relevant baseline as it operates in an *I-ZSL setting*. Standard ZSL methods rely on *image features* for training or adaptation, making them less suitable for image-free environments. For the diagnostic variants marked with *, unavailable image-dependent inputs or stages are replaced or omitted; these variants are not reproductions of the complete original algorithms. No images, stored image features, CLIP image embeddings, generated visual samples, or unseen validation data are used during expansion. When an original method requires image-level validation, we use a fixed published or default value instead. Within each of five runs, methods share the deployed classifier head and fixed feature extractor; trainable methods also use matched adaptation seeds.

F.3 Model Architecture and Implementation Details

Model architecture details. GEORWS follows the same overall design pattern as ICIS: we keep the pre-trained image classifier Φ fixed, and learn a small set of MLP modules that map text-derived semantic signals into mixture coefficients over the seen-class classifier weights. All trainable parts are lightweight compared to the frozen backbone.

We start from a pre-trained classifier Φ with feature extractor $\mathcal{F}$ and a linear classifier head with seen-class weights $\mathbf{W}_{\mathcal{S}} = \{\mathbf{w}_s\}_{s\in\mathcal{S}}$, where each $\mathbf{w}_s \in \mathbb{R}^D$ and $D = 2048$ for the ResNet101 backbone used in our main experiments. During I-ZSL training, we do not use training images, and we do not train $\mathcal{F}$. We also use a frozen LLM to generate class-pair affinity descriptions, and a frozen text encoder f_{ENC} to convert each description into an affinity embedding $\mathbf{a}_{u,s} \in \mathbb{R}^d$, where d is the output dimension of the chosen text encoder (*e.g.*, $d = 512$ for CLIP ViT-B/32 and $d = 768$ for BERT-base/T5-base). The semantic encoder takes a pairwise affinity embedding $\mathbf{a}_{u,s}$ as input and produces a latent representation $\mathbf{z}_{u,s} \in \mathbb{R}^k$. We implement f_θ as a 2-layer MLP with ReLU and hidden width 1000: an input projection $d \to 1000$, followed by ReLU, and an output projection $1000 \to k$. We set $k = 1000$ in all experiments, so $\mathbf{z}_{u,s} \in \mathbb{R}^{1000}$. To compute how much a seen class s should contribute to an unseen class u, we score each pair (u, s) by conditioning on both *(i)* the semantic latent code $\mathbf{z}_{u,s}$ and *(ii)* the seen-class classifier prototype $\mathbf{w}_s$. Concretely, the scoring head takes the concatenation $\mathbf{z}_{u,s} \oplus \mathbf{w}_s \in \mathbb{R}^{k+D}$ and outputs a scalar logit $g_{u,s}$. We implement f_ψ as a 2-layer MLP with ReLU and hidden width 1000: a projection $(k + D) \to 1000$, followed by ReLU, and a projection $1000 \to 1$. The logits are normalized with a softmax over $s \in \mathcal{S}$ to obtain coefficients $\alpha_{u,s}$, which are then used to synthesize the injected unseen-class weight by the convex combination $\mathbf{w}_u = \sum_{s\in\mathcal{S}} \alpha_{u,s}\mathbf{w}_s$. For the semantic reconstruction loss, we include a decoder that maps the latent code back to the text-encoder space. We implement $f_{\tilde{\theta}}$ as a 2-layer MLP with ReLU and hidden width 1000: a projection $k \to 1000$, followed by ReLU, and a projection $1000 \to d$. This decoder is used only for the reconstruction regularizer; it is not used for synthesizing classifier weights. After training, we generate $\mathbf{a}_{u,s}$ for each unseen–seen pair, compute coefficients $\{\alpha_{u,s}\}_{s\in\mathcal{S}}$, synthesize $\mathbf{w}_u$, and inject $\mathbf{W}_{\mathcal{U}}$ into the classifier head. Given a test image x, prediction uses the original feature extractor $\mathcal{F}(x)$ together with the extended classifier weights $\mathbf{W}_{\mathcal{Y}} = \mathbf{W}_{\mathcal{S}} \cup \mathbf{W}_{\mathcal{U}}$, as shown at the end of Section 3.5.

Implementation details. For all methods, we follow a consistent protocol similar to previous work [6]. The visual backbone is ResNet101 [14] pre-trained on ImageNet [8], used for extracting features and predicting seen class labels where applicable. For text embedding, we evaluate CLIP [27], T5 [28], and BERT [9], applying each as specified by the respective baseline. For clarity, the classifier-space (feature/weight) dimension is $D = 2048$ for ResNet101, while the text-encoder output dimension is d (*e.g.*, $d = 512$ for CLIP ViT-B/32 and $d = 768$ for BERT-base/T5-base); our latent dimension is set to $k = 1000$. Training uses a batch size of 16 and a learning rate of 0.00001. All competing methods are re-implemented or adapted using their official code or published configurations, with human-written or LLM-generated descriptions as required. Our approach additionally incorporates a task-adaptive semantic encoder and pairwise affinity module, trained while the deployed classifier remains fixed.

We noticed that some experimental results reported in our paper, such as Table 2, are different from the numbers reported in the original paper. We conduct a fair evaluation in one common environment and using common splits. For each competing method, we integrate its official implementation within our GitHub project: `https://github.com/zhiqiangzhongddu/GeoRWS`.

Computational complexity and scalability. GEORWS builds unseen classifiers from pairwise class relations, leading to a preprocessing complexity of $O(|\mathcal{S}|^2 + |\mathcal{U}||\mathcal{S}|)$ to construct the required affinity embeddings, and a synthesis cost of $O(|\mathcal{U}||\mathcal{S}|D)$ to form unseen weights by weighted sums of seen weights (with D the classifier-space dimension). This pairwise dependence can become a bottleneck when the number of classes grows, and it is an inherent trade-off of using explicit inter-class relations. Despite the pairwise preprocessing term, GEORWS remains computationally light in the learning stage. Training updates only three small MLP modules (f_θ, $f_{\tilde{\theta}}$, f_ψ) while keeping the feature extractor $\mathcal{F}$ and the classifier head frozen. In particular, GEORWS avoids retraining, post-training, or fine-tuning large vision (or vision-language) backbones, which typically dominates compute and storage due to repeated forward/backward passes over large image collections. In our setting, once the affinity embeddings are prepared and cached, model training and inference reduce to

inexpensive MLP forward passes and simple linear combinations of seen-class weights, making the overall pipeline efficient for practical use.

Table 5: Complete ablation study of GEORWS with mean ± standard deviation over five runs. We report ZSL accuracy (Acc), GZSL unseen accuracy (u), seen accuracy (s), and harmonic mean (H). For consistency, H is computed as $H = 2us/(u + s)$.

Settings	Zero-Shot Accuracy			Generalized Zero-Shot Accuracy								
	CUB	AWA2	SUN	CUB			AWA2			SUN		
	Acc	Acc	Acc	u	s	H	u	s	H	u	s	H
GEORWS	**71.28**$_{\pm 0.94}$	**60.34**$_{\pm 1.14}$	**59.07**$_{\pm 0.87}$	**65.05**$_{\pm 0.33}$	**80.08**$_{\pm 0.42}$	**71.79**$_{\pm 0.26}$	37.12$_{\pm 0.63}$	96.21$_{\pm 0.30}$	**53.57**$_{\pm 0.66}$	**48.61**$_{\pm 0.12}$	**45.33**$_{\pm 0.64}$	**46.91**$_{\pm 0.35}$
w/o $\mathcal{L}_{\text{geo}}$	70.55$_{\pm 0.80}$	59.70$_{\pm 1.08}$	58.42$_{\pm 0.86}$	64.20$_{\pm 0.41}$	79.90$_{\pm 0.43}$	71.19$_{\pm 0.30}$	36.08$_{\pm 0.60}$	**96.30**$_{\pm 0.31}$	52.49$_{\pm 0.64}$	47.73$_{\pm 0.15}$	44.98$_{\pm 0.66}$	46.31$_{\pm 0.36}$
w/o $\mathcal{L}_{\text{cons}}$	70.71$_{\pm 0.77}$	59.83$_{\pm 1.02}$	58.55$_{\pm 0.83}$	64.55$_{\pm 0.38}$	79.95$_{\pm 0.41}$	71.43$_{\pm 0.28}$	36.60$_{\pm 0.59}$	96.22$_{\pm 0.32}$	53.03$_{\pm 0.62}$	48.02$_{\pm 0.13}$	45.01$_{\pm 0.65}$	46.47$_{\pm 0.35}$
w/o $\mathcal{L}_{\text{geo}}$ and $\mathcal{L}_{\text{cons}}$	70.39$_{\pm 0.64}$	59.92$_{\pm 0.93}$	58.28$_{\pm 0.87}$	64.29$_{\pm 0.42}$	79.93$_{\pm 0.37}$	71.26$_{\pm 0.30}$	36.40$_{\pm 0.33}$	96.29$_{\pm 0.29}$	52.83$_{\pm 0.35}$	47.76$_{\pm 0.22}$	44.77$_{\pm 0.82}$	46.22$_{\pm 0.45}$
w/o Affinity Description	63.85$_{\pm 0.65}$	53.42$_{\pm 0.89}$	51.27$_{\pm 0.85}$	55.12$_{\pm 0.37}$	78.31$_{\pm 0.35}$	64.70$_{\pm 0.28}$	31.22$_{\pm 0.21}$	94.78$_{\pm 0.16}$	46.97$_{\pm 0.24}$	40.83$_{\pm 0.30}$	42.95$_{\pm 0.77}$	41.86$_{\pm 0.40}$
w/ Class Description	64.23$_{\pm 3.81}$	55.23$_{\pm 2.50}$	57.83$_{\pm 0.92}$	59.00$_{\pm 1.24}$	78.72$_{\pm 0.71}$	67.45$_{\pm 0.85}$	33.23$_{\pm 0.80}$	95.27$_{\pm 1.72}$	49.27$_{\pm 0.91}$	45.52$_{\pm 0.51}$	43.05$_{\pm 1.51}$	44.25$_{\pm 0.83}$
w/o Semantic Encoding	61.74$_{\pm 0.61}$	52.63$_{\pm 0.92}$	49.81$_{\pm 0.84}$	50.48$_{\pm 0.61}$	77.91$_{\pm 0.42}$	61.26$_{\pm 0.47}$	28.37$_{\pm 0.44}$	92.64$_{\pm 0.71}$	43.44$_{\pm 0.52}$	37.55$_{\pm 0.31}$	41.33$_{\pm 0.73}$	39.35$_{\pm 0.37}$
$\lambda = 0.5$	67.89$_{\pm 0.52}$	57.71$_{\pm 0.88}$	55.36$_{\pm 0.58}$	60.41$_{\pm 1.06}$	79.26$_{\pm 0.94}$	68.56$_{\pm 0.77}$	34.61$_{\pm 0.30}$	95.11$_{\pm 0.93}$	50.75$_{\pm 0.35}$	45.92$_{\pm 0.20}$	43.68$_{\pm 0.39}$	44.77$_{\pm 0.23}$
$\lambda = 2$	66.22$_{\pm 0.64}$	56.38$_{\pm 0.82}$	54.19$_{\pm 0.59}$	58.74$_{\pm 0.83}$	78.32$_{\pm 0.92}$	67.13$_{\pm 0.64}$	33.84$_{\pm 0.29}$	96.12$_{\pm 0.22}$	50.06$_{\pm 0.32}$	44.15$_{\pm 0.12}$	41.47$_{\pm 0.53}$	42.77$_{\pm 0.29}$
Only MLLM (GPT-4)	11.37$_{\pm 0.45}$	47.82$_{\pm 0.62}$	26.94$_{\pm 0.57}$	8.42$_{\pm 0.38}$	62.15$_{\pm 0.70}$	14.83$_{\pm 0.59}$	**41.26**$_{\pm 0.59}$	69.37$_{\pm 0.65}$	51.74$_{\pm 0.50}$	18.73$_{\pm 0.44}$	47.62$_{\pm 0.61}$	26.89$_{\pm 0.46}$

Table 6: Additional comparison between GEORWS and baselines when baseline class-wise descriptions are replaced by LLM-generated class-wise descriptions. We report ZSL accuracy (Acc), GZSL unseen accuracy (u), seen accuracy (s), and harmonic mean (H). For consistency, H is computed as $H = 2us/(u + s)$. Methods marked with * are adapted to the image-free setting.

Models	Zero-Shot Accuracy			Generalized Zero-Shot Accuracy								
	CUB	AWA2	SUN	CUB			AWA2			SUN		
	Acc	Acc	Acc	u	s	H	u	s	H	u	s	H
ConSE [22]	39.87$_{\pm 0.79}$	43.12$_{\pm 0.52}$	42.21$_{\pm 0.39}$	0.39$_{\pm 0.03}$	83.32$_{\pm 0.87}$	0.78$_{\pm 0.06}$	2.91$_{\pm 0.13}$	93.84$_{\pm 0.15}$	5.64$_{\pm 0.24}$	0.07$_{\pm 0.00}$	47.10$_{\pm 0.27}$	0.14$_{\pm 0.00}$
COSTA [21]	32.01$_{\pm 1.21}$	44.70$_{\pm 1.63}$	17.12$_{\pm 0.97}$	0.00$_{\pm 0.00}$	84.01$_{\pm 1.70}$	0.00$_{\pm 0.00}$	0.00$_{\pm 0.00}$	93.12$_{\pm 1.72}$	0.00$_{\pm 0.00}$	0.00$_{\pm 0.00}$	49.40$_{\pm 2.04}$	0.00$_{\pm 0.00}$
Sub.Reg.* [1]	49.12$_{\pm 1.58}$	45.20$_{\pm 1.14}$	41.85$_{\pm 1.28}$	0.95$_{\pm 0.41}$	**85.25**$_{\pm 1.98}$	1.88$_{\pm 0.80}$	0.00$_{\pm 0.00}$	92.11$_{\pm 1.25}$	0.00$_{\pm 0.00}$	0.00$_{\pm 0.00}$	**50.15**$_{\pm 1.50}$	0.00$_{\pm 0.00}$
wDAE* [13]	45.90$_{\pm 0.90}$	50.65$_{\pm 1.30}$	43.12$_{\pm 2.53}$	0.71$_{\pm 0.05}$	83.92$_{\pm 1.44}$	1.41$_{\pm 0.10}$	0.00$_{\pm 0.00}$	91.53$_{\pm 2.17}$	0.00$_{\pm 0.00}$	0.00$_{\pm 0.00}$	48.20$_{\pm 1.04}$	0.00$_{\pm 0.00}$
WAvg* [38]	1.82$_{\pm 0.43}$	18.90$_{\pm 0.61}$	1.29$_{\pm 0.53}$	1.59$_{\pm 0.20}$	56.10$_{\pm 2.67}$	3.09$_{\pm 0.38}$	8.12$_{\pm 0.10}$	84.93$_{\pm 2.06}$	14.82$_{\pm 0.17}$	1.17$_{\pm 0.35}$	7.01$_{\pm 0.90}$	2.01$_{\pm 0.52}$
SMO* [38]	43.40$_{\pm 2.48}$	52.01$_{\pm 5.50}$	41.87$_{\pm 2.17}$	36.90$_{\pm 1.48}$	50.23$_{\pm 2.12}$	42.55$_{\pm 1.24}$	35.14$_{\pm 0.82}$	83.02$_{\pm 2.18}$	49.38$_{\pm 0.90}$	39.00$_{\pm 1.18}$	6.22$_{\pm 0.55}$	10.73$_{\pm 0.82}$
ICIS [6]	60.90$_{\pm 1.22}$	20.01$_{\pm 0.11}$	50.12$_{\pm 1.37}$	50.03$_{\pm 1.60}$	66.23$_{\pm 1.63}$	57.00$_{\pm 1.20}$	0.01$_{\pm 0.00}$	93.03$_{\pm 1.93}$	0.02$_{\pm 0.00}$	40.00$_{\pm 0.74}$	26.90$_{\pm 1.16}$	32.17$_{\pm 0.86}$
LaBo [39]	47.95$_{\pm 1.30}$	42.21$_{\pm 1.14}$	46.30$_{\pm 1.32}$	43.12$_{\pm 2.64}$	54.67$_{\pm 2.23}$	48.21$_{\pm 1.86}$	28.75$_{\pm 0.86}$	76.00$_{\pm 2.29}$	41.72$_{\pm 0.97}$	39.42$_{\pm 1.70}$	6.91$_{\pm 0.88}$	11.76$_{\pm 1.28}$
ZSLViT* [5]	61.75$_{\pm 2.30}$	49.21$_{\pm 1.99}$	48.09$_{\pm 2.03}$	50.97$_{\pm 1.90}$	68.40$_{\pm 2.78}$	58.41$_{\pm 1.61}$	7.92$_{\pm 1.37}$	90.02$_{\pm 2.01}$	14.56$_{\pm 2.31}$	38.75$_{\pm 1.93}$	24.12$_{\pm 0.86}$	29.73$_{\pm 0.87}$
CoMC* [19]	53.01$_{\pm 1.82}$	46.70$_{\pm 1.00}$	41.38$_{\pm 1.75}$	8.91$_{\pm 0.46}$	73.11$_{\pm 3.23}$	15.88$_{\pm 0.73}$	2.83$_{\pm 0.10}$	93.01$_{\pm 1.42}$	5.49$_{\pm 0.19}$	32.00$_{\pm 1.76}$	22.41$_{\pm 0.87}$	26.36$_{\pm 0.85}$
DFZSL* [32]	51.23$_{\pm 1.10}$	47.56$_{\pm 0.98}$	44.87$_{\pm 1.21}$	39.41$_{\pm 0.85}$	65.72$_{\pm 1.73}$	49.27$_{\pm 0.82}$	27.33$_{\pm 0.69}$	82.15$_{\pm 1.92}$	41.01$_{\pm 0.81}$	32.14$_{\pm 0.61}$	24.88$_{\pm 0.90}$	28.05$_{\pm 0.62}$
FuDD* [10]	40.03$_{\pm 1.32}$	48.31$_{\pm 1.97}$	37.62$_{\pm 0.44}$	39.12$_{\pm 1.20}$	48.10$_{\pm 1.14}$	43.15$_{\pm 0.86}$	30.90$_{\pm 0.85}$	83.54$_{\pm 1.73}$	45.11$_{\pm 0.94}$	36.01$_{\pm 0.92}$	4.11$_{\pm 0.00}$	7.38$_{\pm 0.02}$
GEORWS	**71.28**$_{\pm 0.94}$	**60.34**$_{\pm 1.14}$	**59.07**$_{\pm 0.87}$	**65.05**$_{\pm 0.33}$	80.08$_{\pm 0.42}$	**71.79**$_{\pm 0.26}$	**37.12**$_{\pm 0.63}$	**96.21**$_{\pm 0.30}$	**53.57**$_{\pm 0.66}$	**48.61**$_{\pm 0.12}$	45.33$_{\pm 0.64}$	**46.91**$_{\pm 0.35}$

G Additional Experimental Results

This section provides additional experimental analyses of GEORWS beyond the main results in Table 2. Unless stated otherwise, all results use the same backbone, class splits, and image-free adaptation protocol as in the main experiments. The complete tables report mean and standard deviation over five random seeds.

G.1 Reporting protocol

For ZSL, we report top-1 accuracy on unseen classes. For GZSL, we report unseen accuracy u, seen accuracy s, and harmonic mean $H = 2us/(u + s)$. When results are averaged over random seeds, we average each metric over runs. Thus, the reported mean H may not exactly equal the harmonic mean computed from the reported mean u and mean s. All standard deviations are computed over five random seeds.

G.2 Hyperparameter selection

The main experiments use $\lambda = 1$ for the semantic reconstruction loss. We select β, γ, and τ_W by minimizing the mean squared seen-class classifier-reconstruction error after optimizing each candidate on the seen-class leave-one-out objective. No images, image-level accuracy, unseen classes, or unseen features are used for selection. The same selected hyperparameters are used for ZSL and GZSL evaluation on each dataset.

For semantic-consistency training, the default augmentation operator $\mathrm{Aug}(\cdot)$ is embedding dropout applied to the affinity embedding $\mathbf{a}_{m,s}$. Description-level perturbations, including text truncation, paraphrasing, and pair swapping, are used only for robustness analysis and are not used to train the main model unless explicitly stated.

G.3 Effect of the geometry and consistency regularizers

Table 3 evaluates the two training regularizers introduced in GEORWS. Removing $\mathcal{L}_{\mathrm{geo}}$ reduces the GZSL harmonic mean from 71.79% to 71.19% on CUB, from 53.57% to 52.49% on AWA2, and from 46.91% to 46.31% on SUN. This indicates that classifier-space neighborhood information helps stabilize the learned affinity coefficients.

Removing $\mathcal{L}_{\mathrm{cons}}$ reduces clean GZSL H by 0.36, 0.54, and 0.44 percentage points on CUB, AWA2, and SUN, respectively. Its main effect is better reflected in the perturbation analysis, where the model is evaluated under noisy or corrupted semantic descriptions. These results suggest that $\mathcal{L}_{\mathrm{geo}}$ mainly improves coefficient alignment under clean inputs, while $\mathcal{L}_{\mathrm{cons}}$ improves robustness under unstable semantic inputs.

The alignment analysis compares learned coefficients with softmax-normalized cosine neighborhoods of seen classifier weights. Pairwise inputs show better neighborhood agreement than class-wise inputs even without $\mathcal{L}_{\mathrm{geo}}$, which further strengthens this agreement. This concerns a mapper trained by classifier reconstruction and conditioned on seen weights; it does not assume that raw frozen-text similarity equals classifier-space similarity.

G.4 Effect of class-pair affinity descriptions

We test whether GEORWS benefits from using *pairwise* LLM descriptions rather than *class-wise* text or label-only inputs. We keep the text encoder, adaptive encoder, scoring head, leave-one-out procedure, classifier weights, optimizer, and optimization settings fixed, and replace only the semantic input. In the class-wise variant, a target-class description from the same LLM is embedded once and reused for every reference; reference-specific information still enters through $\mathbf{w}_s$. This replacement reduces the GZSL harmonic mean by 4.34, 4.30, and 2.66 percentage points on CUB, AWA2, and SUN, respectively, as shown in Table 3. The label-only variant gives larger drops of 7.09, 6.60, and 5.05 points, respectively. These results show that pairwise descriptions provide a stronger transfer signal than independent class-level descriptions or labels alone. These controls isolate an effect beyond using the same LLM for independent descriptions, but do not fully separate all effects of external knowledge and model design.

G.5 Effect of adaptive semantic encoding

We evaluate the role of the adaptive semantic encoder by removing the trainable mapping from text space to the task-aligned latent space. Concretely, we set $\mathbf{z}_{u,s} = \mathbf{a}_{u,s}$, remove the decoder $f_{\tilde{\theta}}$, and keep the same scoring-and-softmax mechanism to produce $\alpha_{u,s}$ from concatenated inputs. This change reduces the GZSL harmonic mean by 10.53, 10.13, and 7.56 percentage points on CUB, AWA2, and SUN, respectively, as shown in Table 3. The result indicates that task-aligned semantic encoding is important for mapping pairwise descriptions into classifier-space affinity coefficients.

G.6 Effect of semantic reconstruction weight

We study the effect of the semantic reconstruction weight λ in Table 3. The default setting $\lambda = 1$ gives the strongest overall performance. Setting $\lambda = 0.5$ weakens semantic reconstruction and reduces the GZSL harmonic mean on all datasets. Setting $\lambda = 2$ gives stronger reconstruction pressure but makes the encoder less flexible for classifier-space alignment. These results support using a balanced objective that preserves semantic information while still allowing task-specific adaptation.

G.7 Robustness to semantic perturbations

Table 4 evaluates robustness to perturbations of pairwise semantic inputs under GZSL. All perturbations are applied at inference time while model parameters are fixed. We evaluate embedding dropout, Gaussian noise in the affinity embedding, text truncation, and random pair swaps. For each unseen–seen pair (u, s), independently with probability ρ, we replace its cached description with that for (u', s), where u' is sampled uniformly from $\mathcal{U} \setminus \{u\}$ and s remains fixed. For each of five matched runs, we average five corruption draws and report standard deviation across the five run-level means. The full GEORWS degrades more slowly than the variant without $\mathcal{L}_{\mathrm{cons}}$, especially under stronger perturbations such as text truncation and pair swapping. This supports the

Table 7: GZSL H under 20% inference-time pair swaps, with mean $\pm$ standard deviation across five run-level means. Training uses embedding dropout rather than pair swapping.

Variant	CUB	AWA2	SUN
GEORWS	68.35 ± 0.58	51.38 ± 0.79	44.40 ± 0.63
w/o $\mathcal{L}_{\text{geo}}$	67.62 ± 0.63	50.58 ± 0.82	43.55 ± 0.69
w/o $\mathcal{L}_{\text{cons}}$	65.70 ± 0.76	49.85 ± 0.84	41.80 ± 0.77
w/o both	64.92 ± 0.83	49.21 ± 0.90	40.96 ± 0.85

role of semantic-consistency regularization in reducing sensitivity to noisy or unstable generated descriptions.

Table 7 separates the two regularizers under 20% corruption. Consistency reduces the clean-to-corrupted decrease in H by $2.29/0.99/2.16$ points on CUB/AWA2/SUN relative to removing $\mathcal{L}_{\text{cons}}$. Random swaps test mismatched relations and do not fully model factual hallucinations, ambiguous names, or systematic domain errors.

For swap rates $0/10/20/40/60\%$, the full model obtains $H = 71.79/70.27/68.35/64.61/60.08$ on CUB, $53.57/52.49/51.38/48.74/45.63$ on AWA2, and $46.91/45.71/44.40/41.57/38.29$ on SUN. Across-run standard deviations are at most $1.12/1.06/0.99$ points, respectively.

G.8 Effect of different LLMs and text encoders

We test alternative LLMs, including GPT-4, Gemini 2.0, and Llama 3, and alternative text encoders, including CLIP, BERT, and T5. Across datasets, the overall ranking is stable: stronger LLMs and stronger text encoders generally yield better class-affinity signals. At the same time, GEORWS remains stable across these choices, suggesting that the adaptive semantic encoder can align different text-derived signals with the classifier space. The results are summarized in Figure 3.

G.9 Transferability of LLM-generated class-wise descriptions

Table 6 evaluates whether replacing baseline class-wise descriptions with LLM-generated class-wise descriptions is sufficient to close the gap to GEORWS. This comparison is not input-matched to GEORWS, because GEORWS uses class-pair descriptions rather than class-wise descriptions. Instead, it tests whether stronger class-wise text alone explains the gains. Most baselines improve only slightly or remain unstable, suggesting that the gains of GEORWS do not come only from stronger text. Rather, the results support the importance of learning classifier-space affinity coefficients through leave-one-out reconstruction and geometry/consistency regularization.

G.10 Classifier and semantic coverage

Since oracle unseen weights are unavailable, we mask seen classes and measure Euclidean distances to the linear span and convex hull of the remaining seen weights. For a scale-independent diagnostic, we normalize each weight before projection; synthesis uses the original weights. Mean span/hull residuals are $0.41 \pm 0.02/0.57 \pm 0.02$ on CUB, $0.48 \pm 0.03/0.63 \pm 0.03$ on AWA2, and $0.36 \pm 0.01/0.52 \pm 0.01$ on SUN. Residuals are averaged over masked classes within each run and then over five runs.

After fixing models and hyperparameters, we measure each masked class's accuracy decrease when its original classifier row is replaced by the learned reconstruction. Spearman correlations across masked classes, averaged over runs, between this decrease and raw/normalized hull residuals are $0.47/0.45$, $0.39/0.37$, and $0.31/0.29$ on CUB/AWA2/SUN; span correlations are $0.34/0.32$, $0.29/0.27$, and $0.23/0.22$ (all across-run standard deviations ≤ 0.06). Maximum frozen-CLIP class-name similarity to the remaining seen classes gives correlations of $-0.43 \pm 0.05/ -0.36 \pm 0.07/ -0.30 \pm 0.04$ with the accuracy decrease. These post-hoc diagnostics use images only after selection and associate weak classifier or semantic coverage with greater accuracy loss; they do not eliminate the coverage limitation.

H Broad Impact and Future Work

This work advances I-ZSL by introducing GEORWS, a framework that eliminates reliance on manually curated semantic descriptions and enables image classification model extension without access to image data. By leveraging LLMs for generating class-pair affinity descriptions and incorporating adaptive text encoding, GEORWS facilitates a new paradigm for transferring knowledge to unseen categories in privacy-sensitive and resource-constrained settings.

The implications of this research extend beyond ZSL, impacting fields where direct data access is restricted due to privacy, security, or ethical concerns. In *healthcare*, GEORWS could enable AI-assisted diagnosis and disease classification across hospitals without sharing patient data, improving accessibility while maintaining compliance with regulatory constraints. In *security*, it provides a means to update recognition models without exposing sensitive training images, mitigating risks associated with data leakage. In *scientific research*, where labeled datasets are often limited, GEORWS offers a scalable approach to classifying novel entities without requiring additional manual annotations.

Beyond its immediate applications, GEORWS contributes to the broader landscape of knowledge transfer and privacy-preserving AI. By demonstrating that ZSL can function effectively without image data, this work opens pathways for more *flexible, generalizable, and ethically responsible* AI systems. Future research could extend these techniques to *multi-modal learning, domain adaptation, and robustness under distribution shifts*, further expanding the applicability of zero-shot classification in real-world environments.

Several limitations remain. First, GEORWS still depends on the quality and task relevance of generated class-pair descriptions. Noisy, incomplete, or biased descriptions may lead to unreliable affinity coefficients, although the semantic-consistency regularizer reduces this risk to some extent. Ambiguous or domain-specific names and systematic factual errors are not fully represented by the random-pair corruption test. Second, the current evaluation focuses on standard I-ZSL benchmarks with fixed label splits. More evidence is needed under larger label spaces, long-tailed taxonomies, stronger domain shift, and evolving class sets. Third, pairwise description generation scales as $O(|\mathcal{S}|^2 + |\mathcal{U}||\mathcal{S}|)$, which is manageable in our experiments but may become costly for very large label sets or frequent label updates. Fourth, because GEORWS uses LLM-generated semantic information, it may inherit biases or factual errors from the underlying LLMs. Careful auditing is needed before using such systems in high-stakes domains.

Finally, the unnormalized synthesized weights lie in the convex hull of seen weights, a stricter restriction than their linear span. Prediction-time normalization changes their scale but does not introduce directions outside the cone generated by the seen weights. Unseen classes with weak semantic or classifier-space coverage may therefore remain difficult to recognize. The learned alignment and consistency constraints characterize these limitations but do not eliminate them or guarantee robustness to arbitrary LLM errors.

Future work includes reducing pairwise generation cost through retrieval or sparse reference selection, improving description calibration and bias filtering, evaluating additional backbones and larger datasets, and incorporating structured domain knowledge such as attributes, ontologies, or class taxonomies. These directions may improve reliability when the semantic relation between seen and unseen labels is weak or when the target domain differs from the source classifier.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions of GeoRWS, including read-only classifier expansion, pairwise semantic descriptions, classifier-space alignment, and geometry/consistency regularization. The claims are supported by the methodology, theory discussion, and experiments in Sections 3 and 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations and future work are discussed in Appendix H. The discussion covers dependence on generated descriptions, possible LLM bias, scalability of pairwise description generation, evaluation scope, and future extensions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The main theoretical statements and assumptions are given in Section 3.6. Full proofs and supporting lemmas are provided in Appendix E.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper specifies datasets, official splits, baselines, adaptation protocol, model architecture, training details, hyperparameters, and evaluation metrics in Appendices F.1–F.3. The algorithm is provided in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide a code and data package for reproducing the main results, including generated descriptions, text embeddings, learned affinity coefficients, and scripts for running GeoRWS and adapted baselines. Instructions for data preparation and reproduction are included in the supplemental material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section F.1 specifies datasets, splits, baselines, and the evaluation protocol. Appendix F.3 provides model architecture, optimizer, batch size, learning rate, text encoders, backbone, hyperparameter selection, and implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Main and appendix tables report mean and standard deviation over five random seeds. The paper states that the reported variation is computed across repeated runs with different random seeds.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix F.3 reports the main compute setup, implementation details, pairwise preprocessing complexity, and LLM query cost. The trainable modules are lightweight MLPs, while the frozen feature extractor and text encoders are not fine-tuned.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses public benchmark datasets, frozen pretrained components, generated semantic descriptions, and standard evaluation protocols. We reviewed the NeurIPS Code of Ethics and do not identify any deviation from it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Appendix H discusses positive impacts such as enabling classifier extension when image data cannot be reused, and negative risks such as inherited LLM bias, incorrect generated descriptions, and possible misuse in sensitive domains.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a new high-risk pretrained model, image generator, scraped dataset, or dual-use system. The released assets are code, generated text descriptions, text embeddings, and classifier-weight synthesis scripts for public benchmark datasets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites the original datasets, baselines, pretrained models, and text encoders used in the experiments. Dataset sources and usage terms are documented in Appendix F.1, and implementation details for reused baselines are provided in Appendix F.2.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release generated class-pair descriptions, text embeddings, learned affinity coefficients, and code for reproducing the experiments. The supplemental package includes documentation, data organization, and reproduction instructions.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: The only human involvement is a small human-written description baseline used for comparison with LLM-generated descriptions. The annotation instructions, task description, participant background, and compensation or volunteer-status details are provided in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: The human-written description comparison involved non-sensitive annotation of public class labels and did not collect personal, biometric, behavioral, or sensitive data. No IRB approval was obtained; the paper describes the low-risk annotation setting and participant instructions in Appendix B.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are a core component of GeoRWS because they generate class-pair affinity descriptions. The paper describes the LLM prompting strategy, model choices, generated descriptions, ablations over different LLMs, and LLM query cost in Sections 3.3 and 4.2 and Appendices A.1–B.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

Algorithm 1 GEORWS: Training and inference under I-ZSL

Require: Seen class set $\mathcal{S}$; unseen class set $\mathcal{U}$; a pre-trained classifier $\Phi : \mathbf{X} \to \mathcal{S}$ with feature extractor $\mathcal{F}$ and seen-class classifier weights $\mathbf{W}_{\mathcal{S}} = \{\mathbf{w}_s\}_{s\in\mathcal{S}}$. During adaptation, no training images or image features are used, and $\mathcal{F}$ is not updated. At test time, $\mathcal{F}$ is used only for prediction with the extended classifier head.

I-ZSL objective: Extend Φ to $\hat{\Phi} : \mathbf{X} \to \mathcal{Y}$, where $\mathcal{Y} = \mathcal{S} \cup \mathcal{U}$, by inferring classifier weights for unseen classes and injecting them into the classifier head.

Input: Frozen LLM f_{LLM}; frozen text encoder f_{ENC}; semantic encoder f_θ with parameters θ; semantic decoder $f_{\tilde{\theta}}$ with parameters $\tilde{\theta}$; scoring head f_ψ with parameters ψ; semantic augmentation operator $\text{Aug}(\cdot)$; geometry temperature τ_W; loss weights λ, β, γ; training steps T.

Phase 1: Training by leave-one-out reconstruction

for $m \in \mathcal{S}$ **do**
  for $s \in \mathcal{S} \setminus \{m\}$ **do**
    $\mathbf{a}_{m,s} \leftarrow f_{\text{ENC}}\big(f_{\text{LLM}}(m, s)\big)$
  end for
end for
for $t = 1$ **to** T **do**
  Initialize $\mathcal{L}_{\text{align}} \leftarrow 0$, $\mathcal{L}_{\text{recon}} \leftarrow 0$, $\mathcal{L}_{\text{geo}} \leftarrow 0$, $\mathcal{L}_{\text{cons}} \leftarrow 0$
  for $m \in \mathcal{S}$ **do**
    for $s \in \mathcal{S} \setminus \{m\}$ **do**
      $\mathbf{z}_{m,s} \leftarrow f_\theta(\mathbf{a}_{m,s})$
      $g_{m,s} \leftarrow f_\psi(\mathbf{z}_{m,s} \oplus \mathbf{w}_s)$
      $\mathcal{L}_{\text{recon}} \leftarrow \mathcal{L}_{\text{recon}} + \|\mathbf{a}_{m,s} - f_{\tilde{\theta}}(\mathbf{z}_{m,s})\|_2^2$
      $\tilde{\mathbf{a}}_{m,s} \leftarrow \text{Aug}(\mathbf{a}_{m,s})$
      $\tilde{\mathbf{z}}_{m,s} \leftarrow f_\theta(\tilde{\mathbf{a}}_{m,s})$
      $\tilde{g}_{m,s} \leftarrow f_\psi(\tilde{\mathbf{z}}_{m,s} \oplus \mathbf{w}_s)$
    end for
    $\alpha_{m,\cdot} \leftarrow \text{softmax}(\{g_{m,s}\}_{s\in\mathcal{S}\setminus\{m\}})$
    $\tilde{\alpha}_{m,\cdot} \leftarrow \text{softmax}(\{\tilde{g}_{m,s}\}_{s\in\mathcal{S}\setminus\{m\}})$
    $\hat{\mathbf{w}}_m \leftarrow \sum_{s\in\mathcal{S}\setminus\{m\}} \alpha_{m,s}\mathbf{w}_s$
    $\tilde{\mathbf{w}}_m \leftarrow \sum_{s\in\mathcal{S}\setminus\{m\}} \tilde{\alpha}_{m,s}\mathbf{w}_s$
    $\mathcal{L}_{\text{align}} \leftarrow \mathcal{L}_{\text{align}} + \|\hat{\mathbf{w}}_m - \mathbf{w}_m\|_2^2$
    Define $p^W_{m,s} = \frac{\exp(\cos(\mathbf{w}_m, \mathbf{w}_s)/\tau_W)}{\sum_{s'\in\mathcal{S}\setminus\{m\}} \exp(\cos(\mathbf{w}_m, \mathbf{w}_{s'})/\tau_W)}$ for $s \in \mathcal{S} \setminus \{m\}$
    $\mathcal{L}_{\text{geo}} \leftarrow \mathcal{L}_{\text{geo}} + \text{KL}\big(p^W_{m,\cdot} \,\|\, \alpha_{m,\cdot}\big)$
    $\mathcal{L}_{\text{cons}} \leftarrow \mathcal{L}_{\text{cons}} + \|\hat{\mathbf{w}}_m - \tilde{\mathbf{w}}_m\|_2^2 + D_{\text{SKL}}(\alpha_{m,\cdot}, \tilde{\alpha}_{m,\cdot})$
  end for
  Update $(\theta, \tilde{\theta}, \psi)$ by minimizing $\mathcal{L} = \mathcal{L}_{\text{align}} + \lambda\mathcal{L}_{\text{recon}} + \beta\mathcal{L}_{\text{geo}} + \gamma\mathcal{L}_{\text{cons}}$.
end for

Phase 2: Inference by classifier injection

for $u \in \mathcal{U}$ **do**
  for $s \in \mathcal{S}$ **do**
    Retrieve cached $\mathbf{a}_{u,s} = f_{\text{ENC}}\big(f_{\text{LLM}}(u, s)\big)$.
    $\mathbf{z}_{u,s} \leftarrow f_\theta(\mathbf{a}_{u,s})$
    $g_{u,s} \leftarrow f_\psi(\mathbf{z}_{u,s} \oplus \mathbf{w}_s)$
  end for
  $\alpha_{u,\cdot} \leftarrow \text{softmax}(\{g_{u,s}\}_{s\in\mathcal{S}})$
  $\mathbf{w}_u \leftarrow \sum_{s\in\mathcal{S}} \alpha_{u,s}\mathbf{w}_s$
end for

Output: Extended classifier $\hat{\Phi}$ with weights $\mathbf{W}_{\mathcal{Y}} = \mathbf{W}_{\mathcal{S}} \cup \mathbf{W}_{\mathcal{U}}, \quad \mathbf{W}_{\mathcal{U}} = \{\mathbf{w}_u\}_{u\in\mathcal{U}}$. Prediction uses the cosine rule in Section 3.5.