Enhancing Zero-Shot Learning: Integrating CLIP Embeddings with Knowledge Graphs and Graph Convolutional Networks

Charith Purushotham, Arjyahi Bhattacharya

Course CSCI 5922, University of Colorado Boulder

Abstract. Zero-shot learning (ZSL) addresses the challenge of recognizing categories without labeled examples, which is essential for scalable and adaptive vision systems. While CLIP models have shown promise in ZSL by aligning images and text in a shared embedding space, they struggle to capture structured inter-class relationships and nuanced semantic dependencies, limiting their generalization to visually similar or semantically complex unseen classes. In this work, we propose a novel multimodal framework that enhances zero-shot classification by fusing CLIP embeddings with structured knowledge graph reasoning and prototype networks. We construct class prototypes by combining CLIP-derived text embeddings with relational embeddings obtained from a Relational Graph Convolutional Network (R-GCN), guided by a prototype refinement loss over a semantically enriched knowledge graph. A lightweight MLP encoder maps CLIP image embeddings into this refined prototype space using mean squared error, and classification is performed via cosine similarity. We evaluate our model on benchmark ZSL datasets, showing that it outperforms baseline CLIP methods and demonstrates the effectiveness of incorporating relational semantics for robust prototype-guided generalization.

Keywords: zero shot learning, knowledge graphs, graph convolutional networks, relational graph convolutional networks, CLIP embeddings, prototype networks, semantic learning, predicate semantics, contrastive relations, multimodal representations, prototype refinement loss

1 Introduction

1.1 Motivation

The ability to recognize and reason about previously unseen categories is a core aspect of human intelligence and a critical capability for scalable machine learning systems. In many domains, such as healthcare diagnostics, ecological monitoring, and industrial inspection, it is infeasible to collect sufficient labeled data for every possible class. Zero-Shot Learning (ZSL), which enables models to classify novel categories without direct supervision, addresses this challenge by relying on auxiliary semantic information. However, most existing approaches either underperform in real-world generalization tasks or fail to incorporate the rich structure of domain knowledge. Developing more robust and semantically aware ZSL models is essential for enabling intelligent systems that can adapt to novel concepts with minimal human intervention.

1.2 Limitations of Existing Work

Recent models such as CLIP have shown promise in zero-shot classification by aligning images and textual descriptions in a shared embedding space. Despite their effectiveness, these models rely primarily on surface-level image-text similarity, which limits their ability to model fine-grained semantic relationships or leverage contextual and hierarchical knowledge among classes. For instance, CLIP may struggle to distinguish between semantically close but visually distinct categories due to its lack of structured reasoning. Although some prior efforts integrate knowledge graphs or class hierarchies, they often use them as isolated modules or post-processing steps rather than deeply fusing them with visual-language representations. Consequently, existing methods do not fully exploit structured knowledge to improve semantic alignment and generalization in ZSL settings.

1.3 Proposed Approach and Contribution

In this work, we propose a novel zero-shot classification framework that integrates CLIP embeddings, structured semantic reasoning via knowledge graphs, and prototype-based matching into a unified system. Our model constructs class prototypes by combining CLIP text embeddings with knowledge-aware graph embeddings derived from a Relational Graph Convolutional Network (R-GCN) trained on a class-level knowledge graph. A lightweight encoder is then trained to map CLIP image embeddings into this enriched semantic space. Classification is performed by computing the similarity between the encoded image and prototype representations. This approach enables the model to reason about unseen classes based on both linguistic and relational context, rather than relying solely on literal text-image alignment. Our key contributions include a novel fusion of visual, textual, and graph-based semantics, and the design of a prototype refinement mechanism guided by structured knowledge- leading to more interpretable and generalizable zero-shot predictions.

2 Related Work

Zero-Shot Learning with Knowledge Graphs [1][2][3][4][5][6] Our work differs from these by integrating CLIP's image-text embeddings with knowledge graphs, enhancing zero-shot image classification through a novel combination of visual and semantic information.

Graph Convolutional Networks (GCNs) in Zero-Shot Learning [7][8][5][6]

Unlike these studies that focus on GCNs for node classification, our approach applies GCNs to enrich CLIP embeddings for zero-shot image classification, leveraging graph structures to enhance visual-semantic alignment.

Prototype-Based Approaches in Zero-Shot Learning [9][10][11][12]

Our method extends prototype-based learning by incorporating GCN-enhanced CLIP embeddings, creating prototypes that benefit from both visual and structured semantic information for improved zero-shot classification.

Adapting CLIP for Zero-Shot Classification [13][14][15]

While these works adapt CLIP for classification tasks, our approach uniquely combines CLIP embeddings with knowledge graphs and GCNs to enhance zero-shot learning, leveraging structured semantic relationships alongside CLIP's capabilities.

Prototype Refinement and Semantic Alignment Losses [9][16][17][18]

Our work introduces a prototype refinement loss that enforces semantic consistency between refined prototypes and original CLIP embeddings, guided by knowledge graph structures, which is distinct from prior methods that do not incorporate such structured semantic supervision.

Zero-Shot Learning with CLIP and Semantic Graphs [14][13][11]

Our approach differs by combining CLIP's image-text embeddings with structured semantic information from knowledge graphs, enriching the embeddings through GCNs to enhance zero-shot classification, as opposed to solely relying on image-text similarity or traditional semantic embeddings.

3 Methodology

Our proposed method enhances zero-shot classification by combining CLIP's text embeddings with structured relational knowledge via a knowledge graph (KG) and graph-based prototype refinement. We further aligned image representations to these refined class prototypes through a lightweight mapping network. This section describes each component in detail.

3.1 Dataset Preparation and Class Splitting

We used the **Animals with Attributes 2 (AWA2)** dataset, which contains labeled images across 50 animal classes along with attribute annotations. We used 10 out of these classes, using 350 images per seen class for training and 20 images per unseen class for testing. Following the zero-shot learning protocol, we split the classes equally into:

- **Seen Classes**: Used during training (e.g., *cow*, *horse*).
- **Unseen Classes**: Used only during testing (e.g., *zebra*, *buffalo*).

This split ensured that unseen class images are never encountered during training, providing a true zero-shot evaluation setting.

3.2 Knowledge Graph Construction

We constructed a **Knowledge Graph (KG)** where nodes represent classes and edges represent semantic relationships. The graph was additionally manually enriched with attribute-based relations as well as contrasts in addition to natural taxonomy-inspired links. Examples of relationships include:

- *Horse* ↔ *Zebra* (both are equines, have hooves).
- *Cow* ↔ *Horse* (both are farm animals).
- *Deer* ↔ *Blue whale* (deer is a land animal while blue whale lives in water).

Node features were initialized using CLIP text embeddings of class descriptions, ensuring compatibility with the downstream image encoder.

3.3 Text Embedding Extraction (CLIP)

For each class c, we created a textual description (e.g., *"a photo of a horse"*) and passed it through the pretrained CLIP text encoder to obtain a fixed-dimensional embedding:

$$\mathbf{t}_c = \text{CLIP}_{\text{text}}(\text{"A photo of a } c\text{"})$$

where $\mathbf{t}_c \in \mathbb{R}^d$ and d is typically 512.

3.4 Prototype Refinement via R-GCN with Predicate-Aware Semantic Graph

To refine the initial CLIP text embeddings of class names using structured semantic relationships, we employ a **Relational Graph Convolutional Network (R-GCN)** trained over a custom knowledge graph $\mathcal{G}_{\text{triples}}$. This graph is composed of predicate-level semantic triples, where nodes represent animal classes and directed edges represent two types of relationships: *(i)* descriptive predicates (e.g., “has_tail”, “has_hooves”), and *(ii)* contrastive predicates (e.g., “contrast_tail”, “contrast_blue”). All relationships are treated as undirected by symmetrically adding reverse-direction triples.

Target Similarity Matrix. To supervise the refinement process, we define a target similarity matrix $\mathbf{S} \in \mathbb{R}^{C\times C}$, where C is the number of classes. For each pair of classes (i, j), a scalar similarity score S_{ij} is computed based on the number of shared and contrasting predicates that were available in the dataset. Let n_{sim} and n_{con} denote the counts of similarity and contrast predicates between class i and j. Then:

$$S_{ij} = \begin{cases} \frac{n_{\text{sim}}+\epsilon}{n_{\text{sim}}+n_{\text{con}}+\epsilon}, & \text{if } n_{\text{con}} < 0.75(n_{\text{sim}} + n_{\text{con}}) \\ 0, & \text{otherwise} \end{cases}$$

where ϵ is a small constant to avoid division by zero. This matrix encourages similar classes to have refined embeddings close to each other, and dissimilar ones to be distant.

R-GCN Training and Prototype Refinement Loss. We train the R-GCN using the semantic graph $\mathcal{G}_{\text{triples}}$ and the target similarity matrix $\mathbf{S}$. The R-GCN refines the initial CLIP-based text embedding $\mathbf{t}_c$ for each class c into a prototype vector $\mathbf{g}_c$ by propagating predicate-level information through the graph:

$$\mathbf{g}_c = \text{R-GCN}(\mathbf{t}_c, \mathcal{G}_{\text{triples}})$$

To ensure that the refined prototypes align with the semantic relationships encoded in the graph, we introduce the **Prototype Refinement Loss** as part of the training objective. The R-GCN is trained for 300 epochs using the Adam optimizer with a learning rate of 10^{-4}.

3.5 Prototype Refinement Loss using Semantic Supervision

To encourage the R-GCN to produce refined class prototypes that preserve both semantic alignment with the original CLIP embeddings and structured dissimilarity based on knowledge graph relationships, we introduce a custom **Prototype Refinement Loss**.

Given a set of refined class embeddings $\mathbf{g} \in \mathbb{R}^{C \times d}$ from the R-GCN and the original CLIP-based embeddings $\mathbf{z} \in \mathbb{R}^{C \times d}$, our loss consists of two components:

1. **Alignment Loss:** This term ensures the refined prototypes stay close to the original ones in the CLIP semantic space. We compute the average cosine similarity between corresponding pairs and define the alignment loss as:

$$\mathcal{L}_{\text{align}} = 1 - \frac{1}{C} \sum_{i=1}^{C} \cos(\mathbf{g}_i, \mathbf{z}_i)$$

2. **Separation Loss:** To enforce inter-class structural relationships, we construct a soft similarity matrix $S \in \mathbb{R}^{C \times C}$ based on the frequency of similar and contrastive predicates between each class pair in the knowledge graph. We compute the pairwise cosine similarity of the refined prototypes and apply a mean squared error loss with respect to S:

$$\mathcal{L}_{\text{sep}} = \frac{1}{C^2} \sum_{i,j} \left(\cos(\mathbf{g}_i, \mathbf{g}_j) - S_{i,j}\right)^2$$

The total prototype refinement loss is a weighted combination of the two terms:

$$\mathcal{L}_{\text{proto}} = \alpha \cdot \mathcal{L}_{\text{align}} + \beta \cdot \mathcal{L}_{\text{sep}}$$

We set $\alpha = 1.0$ and $\beta = 0.8$ based on validation performance. The soft target similarity matrix S is derived by analyzing the KG's edge structure: for each class pair (c_1, c_2), we compute the proportion of "similar" relations out of all predicate types (including contrastive ones). Pairs with overwhelmingly contrastive links are penalized to have zero similarity.

Motivation for Similarity-Based Supervision: Rather than treating class prototypes as independent, we aim to structure their representations to reflect underlying semantic relationships. By supervising the R-GCN with a soft similarity matrix derived from the knowledge graph, we enable it to encode graded proximity between class embeddings. For example, the refined prototype for "horse" should lie closer to "cow" than to "killer whale," based on shared versus contrasting predicates. This encourages the model to learn a semantically meaningful topology over the prototype space, improving its ability to generalize to unseen but related classes (e.g., zebra). The separation loss component ensures this structure is reflected in the learned embeddings, while the alignment loss keeps prototypes faithful to their original CLIP representations.

3.6 Prototype Construction

Once trained, the refined prototypes $\{\mathbf{g}_c\}$ are extracted and reordered according to a predefined canonical order of classes. This allows for consistent downstream comparisons. The reordered prototypes are then saved for further analysis. After refinement, each class c has a final prototype vector:

$$\mathbf{p}_c = \mathbf{g}_c$$

These prototypes are used as targets during the mapping of image embeddings.

3.7 Image Embedding Extraction (CLIP)

Each training image (i.e., images of seen classes) x_i was passed through the frozen CLIP image encoder to obtain an image embedding:

$$\mathbf{v}_i = \text{CLIP}_{\text{image}}(x_i)$$

where $\mathbf{v}_i \in \mathbb{R}^d$.

3.8 Lightweight Mapping Network and Training Objective

To align the CLIP image embeddings with the refined class prototypes, we trained a lightweight Multi-Layer Perceptron (MLP) mapping network f_θ:

$$\hat{\mathbf{v}}_i = f_\theta(\mathbf{v}_i)$$

The MLP is optimized using a Mean Squared Error (MSE) loss between the mapped image embedding $\hat{\mathbf{v}}_i$ and the prototype of its ground-truth class $\mathbf{p}_y$:

$$\mathcal{L}_{\text{map}} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\mathbf{v}}_i - \mathbf{p}_y\|_2^2 \tag{1}$$

where N is the number of training examples, and $\|\cdot\|_2$ denotes the Euclidean norm. This loss encourages the mapped image embeddings to be close to their corresponding class prototypes in the embedding space.

The training process for the MLP can be summarized as follows:

1. **Loading Prototypes:** The class prototypes are loaded from a pre-trained file, where they represent the refined class embeddings:
$$\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_C\}$$
where C is the number of classes.
2. **Loading Image Embeddings:** For each class, the image embeddings are loaded, and the corresponding labels are generated. The image embeddings $\mathbf{v}_i$ are normalized to unit length.
3. **Generating Soft Similarity Targets:** The soft similarity targets are computed by normalizing both the image embeddings $\mathbf{v}_i$ and the prototypes $\mathbf{p}_y$, then computing the cosine similarity between the normalized embeddings:
$$S_{ij} = \frac{\mathbf{v}_i \cdot \mathbf{p}_j}{\|\mathbf{v}_i\|_2 \|\mathbf{p}_j\|_2}$$
4. **Training the MLP:** The MLP is trained with the objective of minimizing the MSE loss between the mapped image embeddings and the soft similarity targets. The loss is computed as:
$$\mathcal{L}_{\text{map}} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\mathbf{v}}_i - \mathbf{p}_y\|_2^2$$
where $\hat{\mathbf{v}}_i$ are the outputs of the MLP.
5. **Optimization:** We use the Adam optimizer with a learning rate of 1×10^{-3} and train the network for 10 epochs. At each epoch, the model is updated using backpropagation to minimize the loss.
6. **Saving the Model:** After each epoch, the model's state is saved to a checkpoint file. The final model is also saved after training.

3.9 Inference Pipeline (Zero-Shot Prediction)

At test time, for an unseen image x_{test}, the following steps were performed:

1. The CLIP image embedding for the test image was extracted:
$$\mathbf{v}_{\text{test}} = \text{CLIP}_{\text{image}}(x_{\text{test}})$$
2. The embedding was mapped via the trained Multi-Layer Perceptron (MLP):
$$\hat{\mathbf{v}}_{\text{test}} = f_\theta(\mathbf{v}_{\text{test}})$$
3. The class with the highest cosine similarity between the mapped embedding and the prototype matrix was predicted:
$$\hat{y} = \arg \max_{c \in \mathcal{C}_{\text{unseen}}} \cos(\hat{\mathbf{v}}_{\text{test}}, \mathbf{p}_c)$$

3.10 Evaluation Procedure

To evaluate the model, we performed the following:

- We loaded the trained model and prototype matrix.
- For each test sample, we calculated the cosine similarity between the mapped image embedding and each of the class prototypes.
- We predicted the class corresponding to the highest cosine similarity score. The top-2 predictions were considered to account for potential semantic equivalence between class names (e.g., ”zebra” and ”horse”).
- We computed both top-1 accuracy (direct class match) and adjusted accuracy (including semantic equivalence or the second-best prediction) to evaluate performance.
- Precision, recall, and F1 score metrics using Top 1 predictions were computed to evaluate the model's performance in a balanced manner across all classes.

3.11 Design Rationale

- **Why CLIP?** CLIP offers strong zero-shot capabilities due to its large-scale pretraining on image-text pairs.
- **Why Knowledge Graph?** It enables structured semantic reasoning by incorporating both attribute-level and relational information between classes.
- **Why R-GCN?** It effectively handles multiple relation types in the knowledge graph, enriching class embeddings in a relation-aware manner.
- **Why Prototype Refinement Loss?** It enforces semantic consistency in the prototype space, aiding better generalization to unseen classes.
- **Why MLP Mapper?** It provides flexibility to adapt CLIP image embeddings to the new prototype space without altering the frozen CLIP encoders.

4 Experiments

We designed a series of three experiments to evaluate the contributions of different components in our model - graph structure, relation types, and prototype refinement loss - toward unseen class classification. In each case, we trained a lightweight MLP classifier over the node embeddings and evaluated on unseen classes.

4.1 Experiment 1: CLIP Embeddings + MLP

Setup: We removed graph processing and trained the MLP directly over raw CLIP text embeddings of class names.

Purpose: This served as a strong baseline to test the zero-shot performance of pretrained CLIP embeddings without any graph-based enhancement.

Evaluation Metrics: We evaluated performance using Top-1 Accuracy, Adjusted Accuracy (Top-2), Macro Precision, Macro Recall, and Macro F1-score.

Results: The model achieved a Top-1 Accuracy of 36%, an Adjusted Accuracy of 100%, Macro Precision of 0.3333, Recall of 0.2000, and F1-score of 0.2276. Class-wise performance:

– **Antelope:** 0% (Adjusted: 100%)
– **Buffalo:** 15% (Adjusted: 100%)
– **Chimpanzee:** 65% (Adjusted: 100%)
– **Killer Whale:** 100% (Adjusted: 100%)
– **Zebra:** 0% (Adjusted: 100%)

Discussion: While CLIP embeddings helped the model achieve perfect Adjusted Accuracy, Top-1 results were modest. This indicates that although CLIP provides rich semantic priors, it lacks the ability to contextualize class semantics relative to one another, which is essential for disambiguating unseen classes in fine-grained classification. Interestingly, classes like "Killer Whale" and "Chimpanzee" achieved high scores, likely because they are semantically distinct in the CLIP space. However, visually or contextually similar classes like "Zebra" and "Antelope" were harder to separate without additional structural context.

4.2 Experiment 2: GCN with Only Edge Weights + MSE Loss

Setup: We trained a GCN using only scalar edge weights between classes, without any notion of relation types. CLIP text embeddings were used as initial node features, and the model was trained using Mean Squared Error (MSE) loss to match CLIP embeddings.

Purpose: This setup served as a basic graph baseline to assess how far simple class similarity (based on edge weights) could take the model without semantic relational structure.

Evaluation Metrics: We evaluated performance using Top-1 Accuracy, Adjusted Accuracy (Top-2), Macro Precision, Macro Recall, and Macro F1-score.

Results: The model achieved a Top-1 Accuracy of 43%, an Adjusted Accuracy of 63%, Macro Precision of 0.3389, Recall of 0.3583, and F1-score of 0.3105. Class-wise performance:

– **Antelope:** 55% (Adjusted: 60%)
– **Buffalo:** 65% (Adjusted: 95%)
– **Chimpanzee:** 0% (Adjusted: 60%)
– **Killer Whale:** 0% (Adjusted: 0%)
– **Zebra:** 95% (Adjusted: 100%)

Discussion: The model performed noticeably better than the CLIP-only baseline on several classes, suggesting that even a simplistic graph structure capturing pairwise similarity is beneficial. However, performance varied significantly across classes, with failures on semantically ambiguous categories. This inconsistency suggests that scalar edge weights alone may be insufficient to guide fine-grained reasoning, particularly in cases where multiple unseen classes are conceptually similar or poorly connected in the graph.

4.3 Experiment 3: R-GCN with Relations and Contrasts + Prototype Refined Loss

Setup: We constructed a full knowledge graph incorporating typed relations and contrast predicates. An R-GCN was trained with the Prototype Refined Loss, and final embeddings were passed through an MLP.

Purpose: This final setup aimed to combine structural relational reasoning with semantic alignment to form interpretable and discriminative class prototypes.

Evaluation Metrics: We evaluated performance using Top-1 Accuracy, Adjusted Accuracy (Top-2), Macro Precision, Macro Recall, and Macro F1-score.

Results: The model achieved a Top-1 Accuracy of 59%, an Adjusted Accuracy of 100%, Macro Precision of 0.5556, Recall of 0.3278, and F1-score of 0.3577. Class-wise performance:

- **Antelope:** 80% (Adjusted: 100%)
- **Buffalo:** 95% (Adjusted: 100%)
- **Chimpanzee:** 5% (Adjusted: 100%)
- **Killer Whale:** 100% (Adjusted: 100%)
- **Zebra:** 15% (Adjusted: 100%)

Discussion: This configuration yielded the best overall performance, particularly in Top-1 accuracy and F1-score. The integration of relation and contrast predicates allowed the R-GCN to learn richer semantic structures that help distinguish between classes, especially when those classes are conceptually close. The Prototype Refined Loss further anchored class embeddings to the CLIP semantic space, improving alignment and interpretability. That said, the model still struggled with low precision on classes like ”Chimpanzee” and ”Zebra,” which may suggest that the predicate coverage or quality for these classes is insufficient or noisy.

4.4 Quantitative Comparison

Summary. The final model combining relational structure, contrastive cues, and prototype refinement (Experiment 3) significantly outperformed all baselines.

Table 1. Performance Comparison Across Experiments

Experiment	**Top-1 Acc**	**Adj. Acc**	**Precision**	**Recall**	**F1**
CLIP + MLP (Expt 1)	0.36	1.00	0.3333	0.2000	0.2276
Edge + MSE Loss (Expt 2)	0.43	0.63	0.3389	0.3583	0.3105
R-GCN + PR Loss (Expt 3)	**0.59**	**1.00**	**0.5556**	**0.3278**	**0.3577**

Results demonstrate that a structured knowledge graph with semantically meaningful predicates enables better generalization to unseen classes, especially when guided by prototype alignment. A consistent trend across experiments is the increasing benefit of structured context: from raw CLIP semantics (Expt 1), to similarity-based reasoning (Expt 2), and finally to full relational and contrastive modeling (Expt 3).

Limitations and Future Work. Despite overall improvement, certain classes (e.g., "Chimpanzee," "Zebra") continued to show low Top-1 accuracy, indicating that predicate quality, density, or coverage may vary significantly across the graph. Another limitation lies in our reliance on hand-constructed or static predicate triples, which may not capture nuances or context-specific meanings. Future work could explore dynamically generated graphs using large language models or incorporate vision-language grounding directly into the edge semantics. Additionally, scaling the model to larger and more diverse class vocabularies, along with ablations on different types of predicates (e.g., spatial vs. behavioral), could offer deeper insights into which semantics most influence generalization to unseen categories.

5 Conclusion

We presented a novel zero-shot classification framework that synergistically integrates CLIP embeddings, knowledge graph-based relational reasoning, and prototype networks to improve generalization to unseen classes. Our key innovation lies in the fusion of multimodal information—textual, visual, and structured semantic—into a unified prototype space. By refining class prototypes using a Relational Graph Convolutional Network (R-GCN) trained with a custom prototype refinement loss, we enforced semantically consistent distances between related and unrelated class representations. This structure-aware prototype space, when paired with a lightweight MLP trained to align CLIP image embeddings using mean squared error loss, enabled accurate classification via prototype matching even for completely unseen classes.

Through a series of ablation experiments, we demonstrated the contribution of each component to zero-shot performance. Notably, incorporating relational semantics through the knowledge graph and the prototype refinement loss significantly improved both classification accuracy and semantic discrimination. Our approach outperformed naive CLIP-based baselines, emphasizing the importance of modeling inter-class relationships and predicate-level semantics in zero-shot tasks.

Future work could explore scaling this framework to larger and diverse datasets, incorporating automatically mined relationships, and extending it to generalized zero-shot settings. We also envision applications beyond object classification, such as cross-modal retrieval and open-world detection, where structured multimodal reasoning remains critical. Overall, this work highlights the potential of combining pretrained vision-language models with structured knowledge to bridge the gap between visual perception and semantic understanding in open-world scenarios.

While our framework enhances semantic generalization in zero-shot settings, it inherits certain limitations from the underlying pretrained models like CLIP. These include potential biases in training data, which may propagate into the knowledge graph embeddings and result in skewed or unfair classifications for underrepresented or culturally specific classes. Furthermore, the use of contrastive predicates raises questions about interpretability and the explainability of learned representations, which are critical in sensitive application domains like healthcare or law enforcement. As zero-shot systems are deployed in increasingly open-world environments, ensuring transparency in how relationships influence predictions and adopting fairness-aware training strategies become essential. Future iterations of this framework should incorporate auditing mechanisms and bias mitigation techniques to address these concerns responsibly.

References

1. Geng, Y., Chen, J., Ye, Z., Yuan, Z., Zhang, W., Chen, H.: Explainable zero-shot learning via attentive graph convolutional network and knowledge graphs. Semantic Web **12**(5) (2021) 741–765
2. Nayak, N.V., Bach, S.H.: Zero-Shot Learning with Common Sense Knowledge Graphs. arXiv e-prints (June 2020) arXiv:2006.10713
3. Cao, W., Wu, Y., Sun, Y., Zhang, H., Ren, J., Gu, D., Wang, X.: A review on multimodal zero-shot learning. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **13**(2) (2023) e1488
4. Chen, J., Geng, Y., Chen, Z., Pan, J.Z., He, Y., Zhang, W., Horrocks, I., Chen, H.: Zero-shot and few-shot learning with knowledge graphs: A comprehensive survey. Proceedings of the IEEE **111**(6) (2023) 653–685
5. Nayak, N.V., Bach, S.H.: Zero-shot learning with common sense knowledge graphs. arXiv preprint arXiv:2006.10713 (2020)
6. Kampffmeyer, M., Chen, Y., Liang, X., Wang, H., Zhang, Y., Xing, E.P.: Rethinking knowledge graph propagation for zero-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2019) 11487–11496
7. Yu, D., Yang, Y., Zhang, R., Wu, Y.: Knowledge embedding based graph convolutional network. In: Proceedings of the web conference 2021. (2021) 1619–1628
8. Wang, Z., Wang, J., Guo, Y., Gong, Z.: Zero-shot node classification with decomposed graph prototype network. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. (2021) 1769–1779

9. Xu, W., Xian, Y., Wang, J., Schiele, B., Akata, Z.: Attribute prototype network for zero-shot learning. Advances in Neural Information Processing Systems **33** (2020) 21969–21980
10. Yu, Y., Ji, Z., Han, J., Zhang, Z.: Episode-based prototype generating network for zero-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2020) 14035–14044
11. Li, X., Ma, J., Yu, J., Xu, T., Zhao, M., Liu, H., Yu, M., Yu, R.: Hapzsl: A hybrid attention prototype network for knowledge graph zero-shot relational learning. Neurocomputing **508** (2022) 324–336
12. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. Advances in neural information processing systems **30** (2017)
13. Zhang, R., Zhang, W., Fang, R., Gao, P., Li, K., Dai, J., Qiao, Y., Li, H.: Tip-adapter: Training-free adaption of clip for few-shot classification. In: European conference on computer vision, Springer (2022) 493–510
14. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning, PmLR (2021) 8748–8763
15. Zhou, Z., Lei, Y., Zhang, B., Liu, L., Liu, Y.: Zegclip: Towards adapting clip for zero-shot semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2023) 11175–11185
16. Jetley, S., Romera-Paredes, B., Jayasumana, S., Torr, P.: Prototypical priors: From improving classification to zero-shot learning. arXiv preprint arXiv:1512.01192 (2015)
17. Qu, H., Wei, J., Shu, X., Wang, W.: Learning clustering-based prototypes for compositional zero-shot learning. arXiv preprint arXiv:2502.06501 (2025)
18. Liu, J., Qin, Y.: Prototype refinement network for few-shot segmentation. arXiv preprint arXiv:2002.03579 (2020)