
TOMCAT : Test-time Comprehensive Knowledge Accumulation for Compositional Zero-Shot Learning

Xudong Yan^{1,2} Songhe Feng^{1,2 *}

¹ School of Computer Science and Technology, Beijing Jiaotong University

² Key Laboratory of Big Data and Artificial Intelligence in Transportation (Beijing Jiaotong University), Ministry of Education

{xud_yan, shfeng}@bjtu.edu.cn

Abstract

Compositional Zero-Shot Learning (CZSL) aims to recognize novel attribute-object compositions based on the knowledge learned from seen ones. Existing methods suffer from performance degradation caused by the distribution shift of label space at test time, which stems from the inclusion of unseen compositions recombined from attributes and objects. To overcome the challenge, we propose a novel approach that accumulates comprehensive knowledge in both textual and visual modalities from unsupervised data to update multimodal prototypes at test time. Building on this, we further design an adaptive update weight to control the degree of prototype adjustment, enabling the model to flexibly adapt to distribution shift during testing. Moreover, a dynamic priority queue is introduced that stores high-confidence images to acquire visual knowledge from historical images for inference. Considering the semantic consistency of multimodal knowledge, we align textual and visual prototypes by multimodal collaborative representation learning. Extensive experiments indicate that our approach achieves state-of-the-art performance on four benchmark datasets under both closed-world and open-world settings. Code will be available at <https://github.com/xud-yan/TOMCAT>.

1 Introduction

Integrating familiar concepts and expanding to novel compositions stand as one of the striking hallmarks of human intelligence [9, 10, 24]. For example, even when confronted with a completely new composition *browned cheese*, humans can swiftly apprehend and recognize it by combining known atomic concepts *browned* and *cheese*. These capacities for compositionality and generalization enable us to infer a wide range of combinations from limited atomic concepts [1], and have spurred the emergence of Compositional Zero-Shot Learning (CZSL) [39, 46, 2, 26, 35]. Specifically, CZSL aims to recognize unseen attribute-object compositions based on the primitive (*i.e.*, attributes and objects) knowledge learned from seen ones [60].

Traditional CZSL approaches either project both images and attribute-object labels into a joint space to learn compositional representations [40, 36, 22], or use two classifiers to predict attributes and objects separately [50, 25, 13]. Recently, benefiting from the impressive multimodal representational abilities of large pre-trained vision-language models (VLMs) like CLIP [49], there have been further significant advancements in CZSL [42, 34, 14, 28]. These methods employ the prompt tuning technique [73], *i.e.*, replacing the hard prompt like "*a photo of [attribute] [object]*" with learnable soft prompt tokens to align images and textual composition labels, thereby fine-tuning VLMs for adaptation to the CZSL task.

*Corresponding author

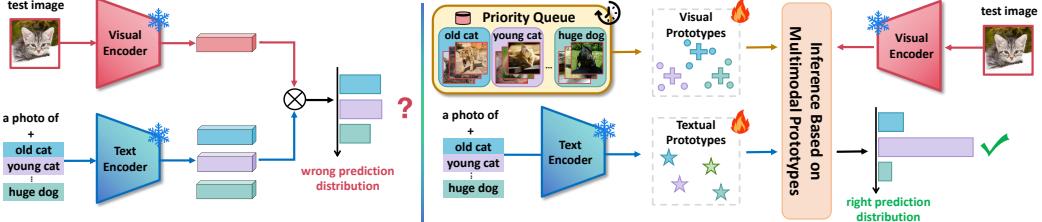


Figure 1: **At test time**, existing methods (*left*) fail to leverage test images for model updates, obtaining wrong prediction distribution caused by label space shift, whereas our TOMCAT (*right*) accumulates multimodal knowledge from unsupervised data at test time to overcome the challenge.

While these pioneering methods have achieved significant progress, they fail to address the performance degradation due to the distribution shift of label space at test time. Specifically, models are assigned to identify unseen attribute-object compositions that are absent during training. This pattern leads to a mismatch between the learned and actual test-time label distribution, bringing about inaccurate predictions and poor generalization. The key reason why the above issue has yet to be effectively addressed lies in the fact that the model parameters and class prototypes are frozen after training, preventing the model from leveraging the test data to adapt towards the new distribution.

However, in more real-world scenarios, an outstanding intelligent system should continuously accumulate knowledge and thus overcome distribution changes after deployment by utilizing the unlabeled data provided through user interactions during the testing phase. In CZSL, the following three aspects should be taken into consideration when using test-time unsupervised data to adjust the model: (1) **Accumulation**. The process of observing test-time images can be regarded as a form of knowledge accumulation, rather than a complete adaptation to unseen compositions at the cost of forgetting knowledge of seen compositions learned during training [23, 43, 58]. (2) **Knowledge comprehensiveness**. Existing methods treat compositions encoded by the text encoder of VLMs as textual-modal prototypes, and predict compositions by computing similarity between the visual features of images and prototypes. However, they overlook the potential of utilizing visual information in the historical images encountered at test time. (3) **Efficiency**. Considering the practical scenario of interaction with users, the method should be time-efficient with low latency [43, 29, 7].

To this end, we propose a **T**est-time **c**OMprehensive knowledge **a**CcumulATion (**TOMCAT**) approach for CZSL, a novel framework that leverages multimodal knowledge of unlabeled data at test time to overcome label distribution shift, as illustrated in Fig. 1. During the test phase, we keep the textual-modal prototypes frozen and progressively learn a Knowledge Accumulation Module (KAM) for prototype adjustment to bridge the label distribution gap with the continual influx of test samples. Subsequently, we determine the extent to which KAM updates the prototypes by employing a designed adaptive update weight strategy, based on similarities between the image and the prototypes. To take full advantage of the visual knowledge from previously seen images, TOMCAT maintains a dynamic priority queue to store high-confidence images for each class. Building on this, visual-modal prototypes are constructed from the images stored in the queue and are dynamically updated—similarly to the textual counterpart—by visual KAM with higher-confidence images enqueued. As testing progresses, the textual and visual prototypes are updated by the entropy minimization objective [54, 52], and are jointly used to facilitate composition recognition under the new label distribution. In addition, given the inherent semantic interdependence between multimodal knowledge, we align textual and visual prototypes by multimodal collaborative representation learning.

Notably, in practical applications where user interaction imposes high requirements on time efficiency, we keep the original model frozen during inference and optimize only the parameters of KAMs via gradient backpropagation once the prediction for a test sample is completed. Meanwhile, extensive experiments on four benchmark datasets (*i.e.*, UT-Zappos [61], MIT-States [15], C-GQA [40], and Clothing16K [68]) demonstrate that our TOMCAT outperforms the state-of-the-art by significant margins in both closed-world and open-world settings (Sec. 3.1).

In summary, the main contributions of our work are three-fold:

- We propose TOMCAT, a novel framework that accumulates multimodal knowledge from unlabeled data and updates prototypes at test time to bridge the label distribution shift. To the best of our knowledge, we are the first to leverage unsupervised data at test time to improve models in CZSL.

- TOMCAT adopts a priority queue that stores historical high-confidence images to calculate visual prototypes and adaptively updates multimodal prototypes by knowledge accumulation modules.
- Extensive experiments conducted on four benchmark datasets demonstrate that our TOMCAT achieves state-of-the-art performance in both closed-world and open-world settings.

2 Related Work

Compositional Zero-Shot Learning (CZSL). CZSL requires the model to recognize unseen compositions with the attribute and object knowledge learned from seen compositions. Previous works in CZSL can be broadly divided into two main streams. One main stream aims at learning representations of compositions by aligning images and textual labels in a shared space, and predicting compositions with the lowest distance [39, 41, 46, 40, 36, 22]. The other stream concentrates on disentangling visual representations of compositions and predicting individual primitives separately to reduce composition learning into primitive learning since both seen and unseen compositions inherently share the same attributes and objects [50, 27, 51, 56, 33, 71]. Rather than learning image-composition association from scratch, recent approaches have increasingly shifted focus to exploiting the multimodal representational capacity of VLMs (*e.g.*, CLIP [49]) for CZSL [42, 34, 14, 28, 3, 59]. For example, Troika [14] exploits multi-path paradigm and cross-modal traction modules to jointly model attributes, objects, and compositions. CDS-CZSL [28] leverages context-based diversity-driven specificity learning to prioritize specific attributes with richer information. ClusPro [47] learns multi-prototypes by within-primitive clustering and dynamically updates them. Although notable progress has been made, these methods freeze model parameters and prototypes after training, which hinders them from using test-time data for further improvement.

Vision-Language Model (VLM). VLMs (*e.g.*, CLIP [49] and ALIGN [16]) pre-trained on web-scale datasets have recently attracted considerable attention due to their impressive capability in aligning visual and textual modalities. Current approaches typically seek to repurpose the multimodal ability of VLMs for various downstream tasks through prompt tuning or adapter tuning. On the text side, prompt tuning is a parameter-efficient technique that replaces manually crafted prompts with a sequence of learnable soft tokens while keeping the text encoder frozen [73, 72, 74]. For example, CoOp [73] eliminates the requirement of manually crafting prompts by prompt tuning under the few-shot setting. CoCoOp [72] extends CoOp by introducing instance-wise conditional prompt learning. On the visual side, adapter tuning refers to injecting lightweight trainable modules (*e.g.*, residual adapters and attention-based adapters) into the visual encoder without fine-tuning the entire backbone [62, 8, 11, 48]. For instance, AdapterFormer [5] augments each Transformer block of the visual backbone with a parallel adapter module and fuses their outputs via element-wise addition. In this work, following CoOp [73] and AdapterFormer [5], we fine-tune CLIP using training data to obtain a simple base model for subsequent testing.

Online Test-time Adaptation (OTTA). OTTA refers to a practical technique where a trained model continually adapts itself by exploiting a stream of unsupervised test data—each test sample is used exactly once, and the model is required to retain and leverage the knowledge gained from earlier test images to improve its performance on later ones in an online manner [54, 55, 44, 32, 4, 6, 30]. For example, Tent [54] proposes to optimize batch normalization layer by minimizing prediction entropy on test data. SAR [44] enhances generalization ability in OTTA by eliminating partially noisy samples with high gradients and promoting the convergence of model weights to a flat minimum. CoTTA [55] is the first work to decrease the accumulation error and avoid catastrophic forgetting through using averaged pseudo-labels and retaining the knowledge of the original model to enhance long-term adaptation. Recently, activating the zero-shot capability of VLMs at test time to mitigate domain shift in downstream tasks has increasingly attracted significant research attention [52, 19, 63, 66, 64, 53]. For instance, TPT [52] develops a prompt tuning approach that learns adaptive prompts on the fly using each test sample with its augmented views. TDA [19] proposes a training-free adapter that uses a lightweight key-value cache and progressive pseudo-label refinement without backpropagation. DynaPrompt [57] adaptively selects and optimizes the relevant prompts for each test built on an online prompt buffer. TPS [53] proposes a straightforward and efficient prototype shifting approach that adjusts per-class prototypes within the embedding space. Notably, OTTA mainly focuses on addressing distribution shifts in the feature domain while we aim to resolve the challenge of label distribution shift caused by unseen compositions recombined from attributes and objects in CZSL. The CZSL task does not have access to any data of unseen compositions during training.

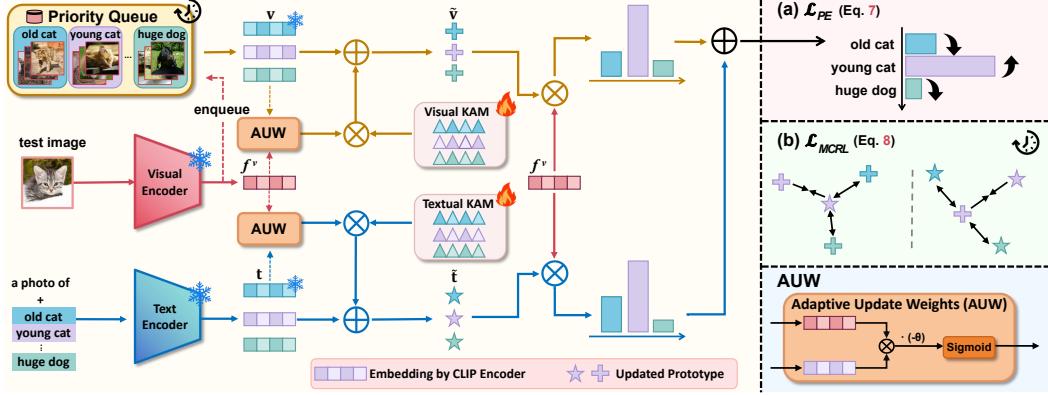


Figure 2: The overall architecture of our proposed TOMCAT at test time. The model accumulates multimodal knowledge to update prototypes to overcome the label distribution shift.

3 Method

3.1 Task Formulation

In this section, we provide a formal description of the CZSL task. Given an attribute set A and an object set O , the composition set can be defined by the Cartesian product of A and O , *i.e.*, $C = A \times O$. And the seen and unseen composition set C^s and C^u are two disjoint subsets of C , *i.e.*, $C^s \cap C^u = \emptyset$. During the training phase, the model learns from a seen training set $D^{tr} = \{(x, c) | x \in \mathcal{X}, c \in C^s\}$, where \mathcal{X} is the image space, and c is the composition label of image x . In the closed-world setting [46], the test composition set C^{te} is defined as the union of C^s and C^u , *i.e.*, $C^{te} = C^s \cup C^u$, where only the presupposed known composition space is considered. For the open-world setting [35], the test composition set expands to all possible attribute-object pairs, *i.e.*, $C^{te} = C$.

3.2 Training Phase

As a preparatory step, we train a simple CLIP-based model by adjusting only its visual and text encoders on the training set, which serves as the foundation for subsequent testing.

Textual Representation Extraction. CLIP treats each attribute-object composition label as a prompt-based textual input, *i.e.*, “*a photo of* [attribute] [object]”, which is subsequently encoded via the text encoder ψ . Given a composition $c = (a, o)$, we create a learnable prompt $\mathbf{P}^c = [\mathbf{p}_1, \dots, \mathbf{p}_l, \mathbf{w}^a, \mathbf{w}^o]$, where $\mathbf{p}_{1:l}$ represents the learnable prompt tokens. \mathbf{w}^a and \mathbf{w}^o are the learnable word tokens of a and o within the attribute and object vocabularies, respectively. Therefore, the textual representation of composition c is obtained, *i.e.*, $f_c^t = \psi(\mathbf{P}^c)$.

Visual Representation Extraction. Given an input image $x \in \mathbb{R}^{H \times W \times 3}$, we feed it into the CLIP visual encoder ϕ and employ the output [CLS] token as its visual representation, *i.e.*, $f_v = \phi(x)$. Based on AdapterFormer [5], a set of learnable adapters is injected into the visual encoder of CLIP, while keeping the initial parameters frozen. Refer to Appendix A for more details.

Training Objective. Given the textual and visual representations, we compute the probability $p(c|x)$ and use the cross-entropy loss to align them:

$$\mathcal{L}_{tr} = -\log p(c|x), \quad p(c|x) = \frac{\exp(\cos(f_v, f_c^t)/\tau)}{\sum_{c' \in C^s} \exp(\cos(f_v, f_{c'}^t)/\tau)}, \quad (1)$$

where τ denotes the temperature parameter of pre-trained CLIP and $\cos(\cdot, \cdot)$ is used to compute cosine similarity. After the training phase, a simple base model is obtained by only tuning adapters and prompt within CLIP, without introducing any additional complex modules to accelerate inference.

3.3 TOMCAT at Test Time

As the major novelty, we introduce TOMCAT to overcome the challenge of label distribution shift by employing unsupervised data at test time, as shown in Fig. 2. Specifically, textual and visual

composition prototypes are obtained by CLIP-encoded textual labels and a dynamic priority queue of historical images, respectively. These multimodal prototypes are then updated by Knowledge Accumulation Modules (KAM) and adaptive update weights, with the objective of minimizing prediction entropy at test time.

Textual Prototype Construction. In line with CLIP’s principles, we treat the embeddings of both seen and unseen composition labels—encoded by the text encoder of the base model after training—as the textual-modal prototypes, *i.e.*, $\mathbf{t} = [\mathbf{t}_{c_1}, \mathbf{t}_{c_2}, \dots, \mathbf{t}_{c_{|C^{te}|}}]^\top \in \mathbb{R}^{|C^{te}| \times d}$.

Visual Prototype Construction. Inspired by TDA [19] and DPE [63], complementary to the text modality, we recognize that prior visual knowledge—captured from historical test images—can be leveraged to further enhance the discriminative capability of CLIP. Therefore, a dynamic priority queue is designed to selectively preserve K ($K = 3$) high-confidence images, which enables TOMCAT to retain representative and reliable exemplars throughout testing. Specifically, for each seen and unseen composition c , we maintain a “confidence-feature” queue, *i.e.*, $q^c = [(h_i, f_i^v)]_{i=1}^K$, where $f_i^v \in \mathbb{R}^d$ is the visual feature of the image x_i in the queue and $h_i \in \mathbb{R}$ is its prediction entropy as confidence:

$$\mathcal{H}(x_i) = - \sum_{c \in C^{te}} p(c|x_i, \tilde{\mathbf{t}}) \log p(c|x_i, \tilde{\mathbf{t}}), \quad p(c|x_i, \tilde{\mathbf{t}}) = \frac{\exp(\cos(f_i^v, \tilde{\mathbf{t}}_c)/\tau)}{\sum_{c' \in C^{te}} \exp(\cos(f_i^v, \tilde{\mathbf{t}}_{c'})/\tau)}, \quad (2)$$

where $\tilde{\mathbf{t}}$ is the updated textual prototypes (defined below in Eq. 5). The lower prediction entropy means higher confidence, and the samples in the queue are sorted by their confidence, *i.e.*, $h_i^c \leq h_{(>i)}^c$.

The priority queue corresponding to each composition is initialized as an empty set \emptyset . For an incoming image x , we first calculate its visual feature f^v and the prediction entropy h , and obtain its pseudo-label c_p by assigning the composition with the highest predicted probability, *i.e.*, $c_p = \arg \max_{c \in C^{te}} p(c|x, \tilde{\mathbf{t}})$. If the queue for c_p is not full, the pair (h, f^v) is directly inserted into the queue. If the queue is full, only if the new image has a prediction entropy lower than the highest element in the queue, the highest one is replaced with (h, f^v) ; otherwise, the queue remains unchanged. Based on the priority queue, the visual-modal prototype for each composition is computed by averaging the visual features, *i.e.*, $\mathbf{v}_c = \frac{1}{K} \sum_{i=1}^K f_i^v$. Subsequently, the visual prototypes are denoted as $\mathbf{v} = [\mathbf{v}_{c_1}, \mathbf{v}_{c_2}, \dots, \mathbf{v}_{c_{|C^{te}|}}]^\top \in \mathbb{R}^{|C^{te}| \times d}$.

Knowledge Accumulation Module (KAM). We aim to continuously acquire information about the new distribution from test samples, while avoiding both catastrophic forgetting of existing knowledge and excessive increases in inference latency. To this end, we introduce learnable KAMs instead of directly modifying the parameters of the base model, and employ adaptive update weights to control the extent to which the KAMs adjust the prototypes. Specifically, for multimodal prototypes \mathbf{t} and \mathbf{v} , multimodal KAMs consist of two sets of learnable parameters, which are initialized to zero:

$$\Delta \mathbf{t} = [\Delta \mathbf{t}_{c_1}, \Delta \mathbf{t}_{c_2}, \dots, \Delta \mathbf{t}_{c_{|C^{te}|}}]^\top \in \mathbb{R}^{|C^{te}| \times d}, \quad \Delta \mathbf{v} = [\Delta \mathbf{v}_{c_1}, \Delta \mathbf{v}_{c_2}, \dots, \Delta \mathbf{v}_{c_{|C^{te}|}}]^\top \in \mathbb{R}^{|C^{te}| \times d}. \quad (3)$$

Adaptive update weight. Taking textual prototypes as an example, we demonstrate the process by which the prototypes are updated with a newly arrived image x . Specifically, the cosine similarity is computed between the visual feature f^v of the image and each original prototype \mathbf{t}_c , based on which the adaptive update weight is calculated as follows:

$$w_c = \sigma(-\theta \cdot s_c), \quad s_c = \cos(f^v, \mathbf{t}_c), \quad (4)$$

where σ denotes the Sigmoid activation function and θ is the hyperparameter that controls the degree of update. Therefore, the updated textual prototypes can be denoted as follows:

$$\tilde{\mathbf{t}} = [\tilde{\mathbf{t}}_{c_1}, \tilde{\mathbf{t}}_{c_2}, \dots, \tilde{\mathbf{t}}_{c_{|C^{te}|}}]^\top, \quad \tilde{\mathbf{t}}_c = \frac{\mathbf{t}_c + w_c \Delta \mathbf{t}_c}{\|\mathbf{t}_c + w_c \Delta \mathbf{t}_c\|}. \quad (5)$$

Accordingly, we can obtain the updated visual prototypes $\tilde{\mathbf{v}}$. This adaptive weighting mechanism enables more controlled updates of the prototypes, thereby avoiding treating all compositions equally regardless of familiarity. Intuitively, when the test image closely matches the original prototype, it is likely associated with a seen composition, and thus excessive adjustments should be avoided. Conversely, a large discrepancy between the test image and the prototype suggests a potentially unseen composition, permitting stronger updates to improve adaptability.

Prediction Entropy Minimization. Following Tip-Adapter [65], the final prediction for the input image is determined as follows:

$$p(c|x, \tilde{\mathbf{t}}, \tilde{\mathbf{v}}) = \frac{\exp(f^v \cdot \tilde{\mathbf{t}}_c + \alpha \mathcal{A}(f^v, \tilde{\mathbf{v}}_c))}{\sum_{c' \in C^{te}} \exp(f^v \cdot \tilde{\mathbf{t}}_{c'} + \alpha \mathcal{A}(f^v, \tilde{\mathbf{v}}_{c'}))}, \quad \mathcal{A}(f^v, \tilde{\mathbf{v}}_c) = \exp(-\beta(1 - f^v \cdot \tilde{\mathbf{v}}_c)), \quad (6)$$

where α and β are hyperparameters controlling multimodal balance and modulating visual-modal sharpness, respectively. At test time, minimizing prediction entropy serves as an unsupervised learning signal that encourages the model to produce more confident predictions in the target label space, thereby enhancing generalization. By reducing uncertainty, the model progressively adjusts to better aligning with the test distribution, which includes unseen compositions. Further explanations are provided in Appendix B. The loss for multimodal prediction entropy is formulated as follows:

$$\mathcal{L}_{PE} = - \sum_{c \in C^{te}} p(c|x, \tilde{\mathbf{t}}_c, \tilde{\mathbf{v}}_c) \log p(c|x, \tilde{\mathbf{t}}_c, \tilde{\mathbf{v}}_c). \quad (7)$$

Multimodal Collaborative Representation Learning. Considering the intrinsic semantic interdependence between multimodal knowledge, we align the textual and visual prototypes by employing multimodal collaborative representation learning. This strategy effectively facilitates the integration of both modalities, thereby enhancing the representation of both textual and visual information in a unified framework. Specifically, contrastive learning is exploited to bring the visual and textual prototypes corresponding to the same composition closer while pushing apart non-corresponding ones, as formulated below:

$$\mathcal{L}_{MCRL} = -\frac{1}{2|C^{te}|} \sum_{c \in C^{te}} \left(\log \frac{\exp(\cos(\tilde{\mathbf{t}}_c, \tilde{\mathbf{v}}_c)/\tau)}{\sum_{c' \in C^{te}} \exp(\cos(\tilde{\mathbf{t}}_c, \tilde{\mathbf{v}}_{c'})/\tau)} + \log \frac{\exp(\cos(\tilde{\mathbf{t}}_c, \tilde{\mathbf{v}}_c)/\tau)}{\sum_{c' \in C^{te}} \exp(\cos(\tilde{\mathbf{t}}_{c'}, \tilde{\mathbf{v}}_c)/\tau)} \right). \quad (8)$$

Testing Pipeline Overview. Upon receiving a test image, we first extract its visual feature and compute the prediction entropy with the original textual prototypes. We then determine whether to update the priority queue of its pseudo-composition label based on the entropy. After performing multimodal prototype refinement by KAMs and adaptive update weights, the final inference prediction for this image is obtained. To reduce latency, the backpropagation update of KAMs is deferred until after the inference step by minimizing the total loss of multimodal prediction entropy and multimodal collaborative representation learning as follows:

$$\mathcal{L}_{TOMCAT} = \mathcal{L}_{PE} + \lambda \mathcal{L}_{MCRL}, \quad (9)$$

where λ is the weighting coefficient of \mathcal{L}_{MCRL} .

4 Experiment

4.1 Experiment Setup

Datasets. Our proposed TOMCAT is evaluated on four commonly used datasets: UT-Zappos [61], MIT-States [15], C-GQA [40], and Clothing16K [68]. UT-Zappos and Clothing16K are two fine-grained fashion datasets, whereas MIT-States and C-GQA consist of images depicting real-world objects. In addition, prior studies [2, 68, 60] have evidenced that MIT-States [15] suffers from considerable noise, with approximately 70% of the labels being incorrect. The detailed introduction and common data splits of the four datasets are presented in Appendix C.

Metrics. Following the evaluation protocol of previous works [46, 35, 42], a bias term from $-\infty$ to $+\infty$ is introduced to trade off the prediction logits between seen and unseen compositions. By varying the bias term, we calculate the best **Seen** accuracy, best **Unseen** accuracy, best Harmonic Mean (**HM**) of seen and unseen accuracies [70], and the Area Under the Curve (**AUC**) drawn with seen and unseen accuracies. In the open-world setting, a post-training feasibility calibration is applied to filter out infeasible compositions within a vast search space [35].

Implementation Details. We implement the base model with CLIP ViT-L/14 architecture in the training phase and TOMCAT at test time in PyTorch [45] framework on a single NVIDIA RTX 3090 GPU. Refer to Appendix D for more implementation details. The source code will also be released at [this website](#) to provide all implementation details and thus facilitate reproducibility.

Baselines. We compare TOMCAT with recent and prominent approaches on UT-Zappos [61], MIT-States [15], and C-GQA [40], including CLIP [49], CoOp [73], CLIP-based Co-CGE [36], DFSP [34],

Table 1: Closed-world and open-world results on UT-Zappos, MIT-States, and C-GQA. The best results are displayed in **boldface**, and the second-best results are underlined. The four indicators are explained in Metrics (Sec. 4.1). In the open-world setting, we report the results of CDS-CZSL* [28] using the same post-training feasibility calibration [35] as our TOMCAT and other baselines use.

Methods	UT-Zappos				MIT-States				C-GQA			
	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen
Closed-world Results												
CLIP [49] (<i>ICML’21</i>)	5.0	15.6	15.8	49.1	11.0	26.1	30.2	46.0	1.4	8.6	7.5	25.0
CoOp [73] (<i>ICCV’22</i>)	18.8	34.6	52.1	49.3	13.5	29.8	34.4	47.6	4.4	17.1	20.5	26.8
Co-CGE [36] (<i>TPAMI’22</i>)	36.3	49.7	63.4	71.3	17.0	33.1	46.7	45.9	5.7	18.9	34.1	21.2
CSP [42] (<i>ICLR’23</i>)	33.0	46.6	64.2	66.2	19.4	36.3	46.6	49.9	6.2	20.5	28.8	26.8
DFSP [34] (<i>CVPR’23</i>)	36.0	47.2	66.7	71.7	20.6	37.3	46.9	52.0	10.5	27.1	38.2	32.0
GIPCOL [59] (<i>WACV’24</i>)	36.2	48.8	65.0	68.5	19.9	36.6	48.5	49.6	7.1	22.5	31.9	28.4
Troika [14] (<i>CVPR’24</i>)	41.7	54.6	66.8	73.8	22.1	39.3	49.0	53.0	12.4	29.4	41.0	35.7
CDS-CZSL [28] (<i>CVPR’24</i>)	39.5	52.7	63.9	<u>74.8</u>	22.4	39.2	<u>50.3</u>	52.9	11.1	28.1	38.3	34.2
PLID [3] (<i>ECCV’24</i>)	38.7	52.4	67.3	68.8	22.1	39.0	49.7	52.4	11.0	27.9	38.8	33.0
IMAX [17] (<i>TPAMI’25</i>)	40.6	54.2	69.3	70.7	21.9	39.1	48.7	<u>53.8</u>	12.8	29.8	39.7	35.8
ClusPro [47] (<i>ICLR’25</i>)	46.6	58.5	70.7	76.0	23.8	40.7	52.1	54.0	14.9	32.8	44.3	37.8
TOMCAT (Ours)	48.3	60.2	74.5	72.8	<u>22.6</u>	<u>39.5</u>	50.3	53.0	16.0	34.0	45.3	40.1
Open-world Results												
CLIP [49] (<i>ICML’21</i>)	2.2	11.2	15.7	20.6	3.0	12.8	30.1	14.3	0.3	4.0	7.5	4.6
CoOp [73] (<i>ICCV’22</i>)	13.2	28.9	52.1	31.5	2.8	12.3	34.6	9.3	0.7	5.5	21.0	4.6
Co-CGE [36] (<i>TPAMI’22</i>)	28.4	45.3	59.9	56.2	5.6	17.7	38.1	20.0	0.9	5.3	33.2	3.9
CSP [42] (<i>ICLR’23</i>)	22.7	38.9	64.1	44.1	5.7	17.4	46.3	15.7	1.2	6.9	28.7	5.2
DFSP [34] (<i>CVPR’23</i>)	30.3	44.0	66.8	60.0	6.8	19.3	47.5	18.5	2.4	10.4	38.3	7.2
GIPCOL [59] (<i>WACV’24</i>)	23.5	40.1	65.0	45.0	6.3	17.9	48.5	16.0	1.3	7.3	31.6	5.5
Troika [14] (<i>CVPR’24</i>)	33.0	47.8	66.4	61.2	7.2	20.1	48.8	18.7	2.7	10.9	40.8	7.9
CDS-CZSL* [28] (<i>CVPR’24</i>)	32.1	48.0	64.7	60.4	—	—	—	—	2.6	10.9	38.2	8.0
PLID [3] (<i>ECCV’24</i>)	30.8	46.6	67.6	55.5	7.3	20.4	49.1	18.7	2.5	10.6	39.1	7.5
IMAX [17] (<i>TPAMI’25</i>)	32.3	47.5	68.4	57.3	7.6	21.4	<u>50.2</u>	18.6	2.6	11.2	38.7	7.9
ClusPro [47] (<i>ICLR’25</i>)	39.5	54.1	71.0	66.2	9.3	23.0	51.2	22.1	3.0	11.6	41.6	8.3
TOMCAT (Ours)	43.7	57.9	74.1	65.8	8.2	21.7	49.2	21.0	4.2	14.2	45.1	10.6

Table 2: Closed-world and open-world results on Clothing16K.

Dataset	Methods	Closed-World				Open-World			
		AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen
Clothing16K									
SymNet [26] (<i>CVPR’20</i>)	78.8	79.3	98.0	85.1	57.4	68.3	98.2	60.7	
CompCos [35] (<i>CVPR’21</i>)	90.3	87.2	98.5	96.8	64.1	70.8	98.2	69.8	
CGE [40] (<i>CVPR’21</i>)	89.2	84.2	98.0	97.4	62.0	68.3	98.5	69.7	
Co-CGE [36] (<i>TPAMI’22</i>)	88.3	87.9	98.5	94.7	59.3	69.2	98.7	63.8	
SCEN [25] (<i>CVPR’22</i>)	78.8	78.5	98.0	89.6	53.7	61.5	96.7	62.3	
INV [68] (<i>ECCV’22</i>)	90.6	86.6	99.0	97.0	63.6	72.0	98.7	69.0	
OADis [51] (<i>CVPR’22</i>)	88.4	86.1	97.7	94.2	53.4	63.2	98.0	58.6	
ADE [13] (<i>CVPR’23</i>)	92.4	88.7	98.2	97.7	68.0	74.2	99.0	73.1	
CLPS [18] (<i>TMM’25</i>)	96.2	91.7	99.2	98.9	71.5	76.1	99.2	76.1	
ATIF [38] (<i>TMM’25</i>)	96.6	95.4	99.0	99.4	<u>91.8</u>	<u>92.0</u>	99.2	<u>93.9</u>	
AIF [38] (<i>TMM’25</i>)	98.2	<u>96.3</u>	99.5	99.6	89.9	90.3	99.7	92.2	
TOMCAT (Ours)	99.5	98.4	99.9	100	95.8	95.0	100	96.4	

GIPCOL [59], Troika [14], CDS-CZSL [28], PLID [3], IMAX [17], and ClusPro [47]. TOMCAT is also compared with some canonical methods that conduct their experiments on Clothing16K [68], including SymNet [26], CompCos [35], CGE [40], Co-CGE [36], SCEN [25], INV [68], OADis [51], ADE [13], CLPS [18], ATIF [38], and AIF [38].

4.2 Main Results

Closed-world results are presented in Table 1 and Table 2. For two fine-grained fashion datasets, our TOMCAT outperforms previous state-of-the-art methods with the improvement of AUC by 1.7%, 1.3%, and HM by 1.7%, 2.1% respectively on UT-Zappos and Clothing16K. Our model also surpasses baselines by 1.1% in AUC and 1.2% in HM on C-GQA. However, TOMCAT falls behind ClusPro and achieves second-best performance on MIT-States, which may be attributed to the substantial

Table 3: Abaltion study of our proposed modules on UT-Zappos and MIT-States. Queue means visual priority queue. T- and V- KAM denote textual and visual KAM. AUW is adaptive update weights.

Queue	Module			UT-Zappos				MIT-States			
	T-KAM	V-KAM	AUW	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen
✓				43.57	55.54	68.72	74.30	22.12	38.97	49.50	52.61
	✓			43.28	55.54	68.43	74.14	22.12	39.15	49.45	52.88
✓		✓		45.68	58.22	74.39	69.01	22.18	39.22	49.50	52.95
✓	✓			43.28	55.54	68.43	74.14	22.32	39.32	49.54	52.93
✓	✓	✓		46.64	58.83	76.44	68.38	22.34	39.39	49.58	52.93
✓	✓	✓	✓	48.31	60.18	74.99	72.77	22.55	39.45	50.32	52.95

Table 4: Ablation study of our designed loss on UT-Zappos and MIT-States.

\mathcal{L}_{PE}	\mathcal{L}_{MCRL}	Loss		UT-Zappos		MIT-States	
		AUC	HM	AUC	HM	AUC	HM
✓		43.57	55.54	22.12	38.97		
	✓	44.59	57.29	22.35	39.32		
✓	✓	42.46	53.97	22.29	39.42		
✓	✓	48.31	60.18	22.55	39.45		

Table 5: Influence of initialization strategies of KAMs on UT-Zappos and MIT-States.

Initialization	UT-Zappos		MIT-States	
	AUC	HM	AUC	HM
Uniform Random	43.49	56.23	21.81	38.59
Normal Random	45.50	57.98	21.32	38.36
Random Walking	47.08	59.12	22.14	39.16
All Zeros	48.31	60.18	22.55	39.45

Table 6: Influence of test order on UT-Zappos, MIT-States, and C-GQA.

Test Order	UT-Zappos				MIT-States				C-GQA			
	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen
Order 1	48.31	60.18	74.49	72.77	22.55	39.45	50.32	52.95	15.98	33.95	45.30	40.12
Order 2	48.52	60.82	74.64	72.95	22.46	39.33	49.84	53.05	15.71	33.75	45.02	39.95
Order 3	46.44	58.72	72.63	72.03	22.31	39.02	49.62	53.09	15.86	34.17	45.42	39.35

noise present in the dataset, leading to inaccurate label supervision and hindering the learning of discriminative features.

Open-world results are shown in Table 1 and Table 2. Our method performs much better than the second-best method by 4.2%, 1.2%, 4.0% in AUC, and 3.8%, 2.6%, 3.0% in HM on UT-Zappos, C-GQA, and Clothing16K, respectively, which indicates that TOMCAT has great potential in the open-world setting.

Discussion. TOMCAT achieves state-of-the-art performance in both closed-world and open-world settings, demonstrating that our proposed method enables the model to adapt to the distribution shift of label space caused by unseen compositions during testing. Particularly in the open-world setting, TOMCAT shows a significant improvement. This can be attributed to the fact that all test images are derived from feasible compositions (both seen and unseen ones), which reinforces our model’s focus on the feasible compositions and reduces its attention to infeasible ones.

4.3 Ablation Study

In this section, we conduct extensive ablation studies to evaluate the contribution of each component within our proposed TOMCAT on UT-Zappos, MIT-States, and C-GQA in the closed-world setting.

Ablation Study of Main Modules. According to Table 3, for MIT-States, empirical results indicate that all proposed modules contribute to the performance improvement of TOMCAT, including the priority queue, multimodal KAMs, and the adaptive update weights. However, on UT-Zappos, incorporating the priority queue and the subsequent visual KAM leads to performance degradation compared to the base model. This phenomenon is attributed to the uniform style of shoes and the subtle inter-composition difference in this dataset, which makes the model insufficient to capture fine-grained semantic distinctions. In contrast, when textual KAM is added, it enables targeted refinement of the textual prototypes based on the stored images. Ultimately, the best performance is

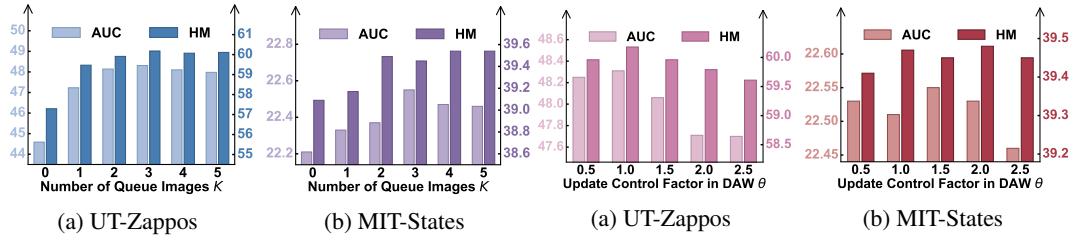


Figure 3: Influence of the number of images in each priority queue K on two datasets.

Figure 4: Influence of the value of update control factor θ on two datasets.

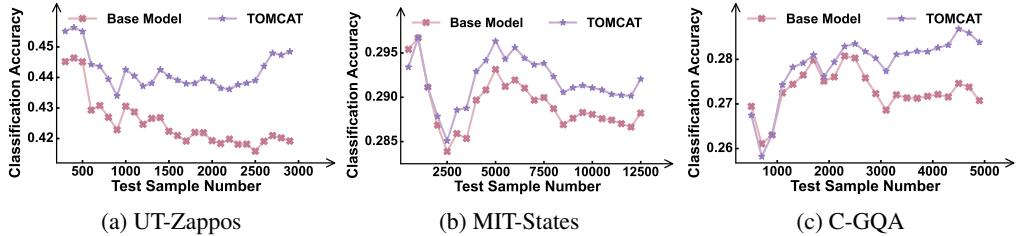


Figure 5: Trend of top-1 classification accuracy with increasing test sample size on three datasets.

achieved by TOMCAT with all modules, which indicates that all components are beneficial and that the multimodal KAMs and priority queue offer complementary advantages.

Ablation Study of Each Loss. As shown in Table 4, the improvement of adding \mathcal{L}_{PE} to the base model indicates that prediction entropy loss helps the model to adapt towards the new label distribution of seen and unseen compositions. However, using only the constraint term \mathcal{L}_{MCRL} may cause the model to blindly reduce the distance between multimodal prototypes without considering other factors, thereby degrading the model’s performance. Finally, by combining \mathcal{L}_{PE} and \mathcal{L}_{MCRL} , TOMCAT achieves the best performance, which confirms the effectiveness of our algorithmic designs.

Influence of Different Initialization Strategies of Multimodal KAMs. Table 5 shows that the zero-initialization outperforms other random initialization strategies. The reason is that starting with random initialization at test-time causes the model to forget the knowledge gained during training.

Influence of Test Order. Since TOMCAT continually accumulates knowledge during testing, the test order may affect the results. We conduct three experiments with different random seeds to vary the sample order. The results in Table 6 suggest that there exists performance variance among different orders, although the differences are not statistically significant.

Influence of Hyperparameters. In Fig. 3 and Fig. 4, we explore the influence of the number of images stored in priority queue K and the update control factor θ respectively on UT-Zappos and MIT-States. TOMCAT achieves the best performance when $K = 3$ on both datasets, as a small number of images causes instability in the visual prototypes, while a large number may include low-confidence images. Moreover, as θ increases, the performance initially improves but subsequently deteriorates. Analysis reveals a small θ leads to insufficient adaptation of the prototypes when encountering images with substantial variation, whereas a large one results in over-updating of the prototypes.

4.4 Qualitative Analysis

In this section, we conduct a qualitative analysis of TOMCAT to visualize its predictions on UT-Zappos, MIT-States, and C-GQA in the closed-world setting.

Trend of Classification Accuracy. To verify whether TOMCAT continuously improves, we visualize in Fig. 5 how top-1 classification accuracy evolves with the number of test samples on three datasets. At the beginning, TOMCAT performs comparably to the base model; however, as more test samples arrive, TOMCAT progressively demonstrates its superiority. The performance gap suggests TOMCAT continuously accumulates useful knowledge to overcome the distribution shift of the label space.



Figure 6: Case study on UT-Zappos and MIT-States. We compare TOMCAT (Ours) with the base model (BM) after training. The successful and failure results are marked in green and red, respectively.

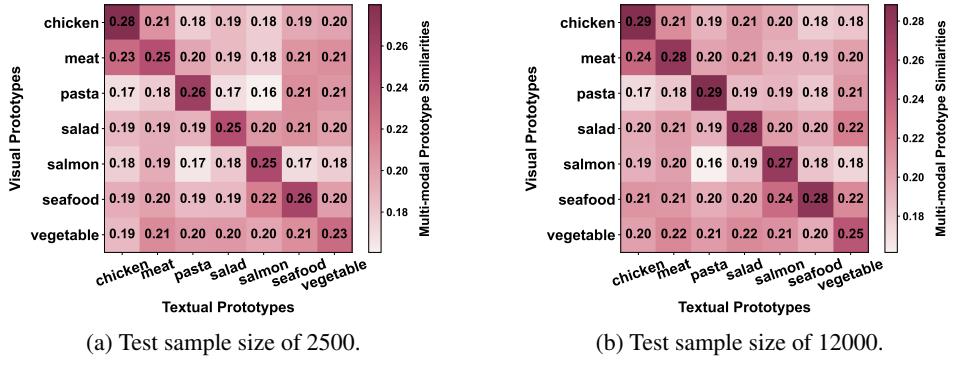


Figure 7: Similarity heatmap of multimodal prototypes on MIT-States. All unseen compositions consisting of the attribute cooked and its corresponding objects (e.g., chicken, meat...) are selected.

Successful and Failure Cases. In Fig. 6, we report top-1 predictions for some image cases. It can be observed that benefiting from adapting to the new distribution, TOMCAT shows superior performance with higher accuracy. For the failure cases, the predictions of TOMCAT can also describe them; for instance, the image of `satin sandals` can also be categorized as `satin heels`.

Similarity Heatmap of Multimodal Prototypes. We visualize the cosine similarities between multimodal prototypes of all unseen compositions consisting of the attribute cooked and its corresponding objects in Fig. 7. Compared to the test size of 2500, the diagonal elements become larger at the test size of 12000, indicating that TOMCAT enhances the connection between multimodal information. Meanwhile, the increase in off-diagonal elements is attributed to all used compositions involving cooked food (almost in the same cluster), where larger similarities can better characterize their semantic structure.

5 Conclusion

In this work, we consider the issue of label distribution shift in CZSL, which arises from novel compositions recomposed from attributes and objects. Therefore, we propose TOMCAT to accumulate comprehensive knowledge of visual and textual modalities from unsupervised data at test time to overcome the challenge. Specifically, we update textual and visual prototypes by multimodal knowledge accumulation modules, and use adaptive update weights to control the degree. Meanwhile, a priority queue is leveraged that stores historical images to obtain the visual knowledge. We hope that our work will inspire further research into exploring label space shift at test time in CZSL.

Limitation. While our proposed TOMCAT accumulates knowledge from test samples to overcome distribution shift of label space at test time, two potential limitations are identified: (1) TOMCAT accumulates comprehensive knowledge only at the level of compositions, neglecting to exploit the information contained within each primitive. (2) Without proper initialization of the priority queue, the model is prone to bias toward the compositions of images already stored during the testing phase.

Acknowledgements

This work was supported by the Fundamental Research Funds for the Central Universities (2025YJS052 and 2025JBZX059), the Beijing Natural Science Foundation (4242046), and the Hebei Province Natural Science Foundation (F2025105018).

References

- [1] Yuval Atzmon, Jonathan Berant, Vahid Kezami, Amir Globerson, and Gal Chechik. Learning to generalize to new compositions in image understanding. *arXiv preprint arXiv:1608.07639*, 2016. 1
- [2] Yuval Atzmon, Felix Kreuk, Uri Shalit, and Gal Chechik. A causal view of compositional zero-shot recognition. In *Advances in Neural Information Processing Systems*, volume 33, pages 1462–1473, 2020. 1, 6, 17
- [3] Wentao Bao, Lichang Chen, Heng Huang, and Yu Kong. Prompting language-informed distribution for compositional zero-shot learning. In *Proceedings of the European Conference on Computer Vision*, pages 107–123, 2024. 3, 7, 22
- [4] Haoran Chen, Micah Goldblum, Zuxuan Wu, and Yu-Gang Jiang. Adaptive retention & correction: Test-time training for continual learning. In *Proceedings of the International Conference on Learning Representations*, 2025. 3
- [5] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. 3, 4, 16, 21
- [6] Zhixiang Chi, Li Gu, Huan Liu, Ziqiang Wang, Yanan Wu, Yang Wang, and Konstantinos N Plataniotis. Learning to adapt frozen clip for few-shot test-time domain adaptation. In *Proceedings of the International Conference on Learning Representations*, 2025. 3
- [7] Zhixiang Chi, Yanan Wu, Li Gu, Huan Liu, Ziqiang Wang, Yang Zhang, Yang Wang, and Konstantinos N Plataniotis. Plug-in feedback self-adaptive attention in clip for training-free open-vocabulary segmentation. *arXiv preprint arXiv:2508.20265*, 2025. 2
- [8] Sanjoy Chowdhury, Sayan Nag, and Dinesh Manocha. Apollo: Unified adapter and prompt learning for vision language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 10173–10187, 2023. 3
- [9] Gottlob Frege. Sense and reference. *The Philosophical Review*, 57(3):209–230, 1948. 1
- [10] Liane Gabora and Kirsty Kitto. *Concept Combination and the Origins of Complex Cognition*, pages 361–381. Springer Netherlands, 2013. 1
- [11] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024. 3
- [12] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, pages 529—536, 2004. 16
- [13] Shaozhe Hao, Kai Han, and Kwan-Yee K. Wong. Learning attention as disentangler for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15315–15324, 2023. 1, 7, 18, 22
- [14] Siteng Huang, Biao Gong, Yutong Feng, Min Zhang, Yiliang Lv, and Donglin Wang. Troika: Multi-path cross-modal traction for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24005–24014, 2024. 1, 3, 7, 21, 22
- [15] Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1383–1391, 2015. 2, 6, 17, 18, 21
- [16] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the International Conference on Machine Learning*, pages 4904–4916, 2021. 3

[17] Chenyi Jiang, Shidong Wang, Yang Long, Zechao Li, Haofeng Zhang, and Ling Shao. Imaginary-connected embedding in complex space for unseen attribute-object discrimination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):1395–1413, 2025. 7, 22

[18] Han Jiang, Chaofan Chen, Xiaoshan Yang, and Changsheng Xu. Compact latent primitive space learning for compositional zero-shot learning. *IEEE Transactions on Multimedia*, 27:4297–4308, 2025. 7, 18, 22

[19] Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14162–14171, 2024. 3, 5, 18, 19, 21

[20] Shyamgopal Karthik, Massimiliano Mancini, and Zeynep Akata. Revisiting visual product for compositional zero-shot learning. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021. 18, 22

[21] Shyamgopal Karthik, Massimiliano Mancini, and Zeynep Akata. Kg-sp: Knowledge guided simple primitives for open world compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9336–9345, 2022. 18, 22

[22] Muhammad Gul Zain Ali Khan, Muhammad Ferjad Naeem, Luc Van Gool, Alain Pagani, Didier Stricker, and Muhammad Zeshan Afzal. Learning attention propagation for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3828–3837, 2023. 1, 3

[23] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 2

[24] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017. 1

[25] Xiangyu Li, Xu Yang, Kun Wei, Cheng Deng, and Muli Yang. Siamese contrastive embedding network for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9326–9335, 2022. 1, 7, 18, 22

[26] Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. Symmetry and group in attribute-object compositions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11313–11322, 2020. 1, 7, 18, 22

[27] Yong-Lu Li, Yue Xu, Xinyu Xu, Xiaohan Mao, and Cewu Lu. Learning single/multi-attribute of object with symmetry and group. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9043–9055, 2022. 3

[28] Yun Li, Zhe Liu, Hang Chen, and Lina Yao. Context-based and diversity-driven specificity in compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17037–17046, 2024. 1, 3, 7, 22

[29] Yushu Li, Xun Xu, Yongyi Su, and Kui Jia. On the robustness of open-world test-time training: Self-training with dynamic prototype expansion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11836–11846, 2023. 2

[30] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025. 3

[31] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 18, 19

[32] Jiaming Liu, Senqiao Yang, Peidong Jia, Renrui Zhang, Ming Lu, Yandong Guo, Wei Xue, and Shanghang Zhang. Vida: Homeostatic visual domain adapter for continual test time adaptation. In *Proceedings of the International Conference on Learning Representations*, 2024. 3

[33] Zhe Liu, Yun Li, Lina Yao, Xiaojun Chang, Wei Fang, Xiaojun Wu, and Abdulmotaleb El Saddik. Simple primitives with feasibility- and contextuality-dependence for open-world compositional zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):543–560, 2024. 3, 18, 22

[34] Xiaocheng Lu, Song Guo, Ziming Liu, and Jingcai Guo. Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23560–23569, 2023. 1, 3, 6, 7, 22

[35] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Open world compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5222–5230, 2021. 1, 4, 6, 7, 22

[36] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Learning graph embeddings for open world compositional zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(3):1545–1560, 2022. 1, 3, 6, 7, 22

[37] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. In *Proceedings of the International Conference on Learning Representations*, 2018. 18

[38] Lintong Min, Ziman Fan, Shunzhou Wang, Feiyang Dou, Xin Li, and Binglu Wang. Adaptive fusion learning for compositional zero-shot recognition. *IEEE Transactions on Multimedia*, 27:1193–1204, 2025. 7

[39] Ishan Misra, Abhinav Gupta, and Martial Hebert. From red wine to red tomato: Composition with context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1160–1169, 2017. 1, 3, 18, 22

[40] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 953–962, 2021. 1, 2, 3, 6, 7, 17, 18, 21

[41] Tushar Nagarajan and Kristen Grauman. Attributes as operators: Factorizing unseen attribute-object compositions. In *Proceedings of the European Conference on Computer Vision*, pages 169–185, 2018. 3, 18, 22

[42] Nihal V. Nayak, Peilin Yu, and Stephen H. Bach. Learning to compose soft prompts for compositional zero-shot learning. In *Proceedings of the International Conference on Learning Representations*, 2023. 1, 3, 6, 7, 22

[43] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofu Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning*, volume 162, pages 16888–16905, 2022. 2

[44] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiqian Wen, Yaofu Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *Proceedings of the International Conference on Learning Representations*, 2023. 3

[45] Adam Paszke, Sam Gross, Francisco Massa, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 6, 18

[46] Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. Task-driven modular networks for zero-shot compositional learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3592–3601, 2019. 1, 3, 4, 6, 18, 22

[47] Hongyu Qu, Jianan Wei, Xiangbo Shu, and Wenguan Wang. Learning clustering-based prototypes for compositional zero-shot learning. In *Proceedings of the International Conference on Learning Representations*, 2025. 3, 7, 22

[48] Hongyu Qu, Rui Yan, Xiangbo Shu, Hailiang Gao, Peng Huang, and Guo-Sen Xie. Mvp-shot: Multi-velocity progressive-alignment framework for few-shot action recognition. *IEEE Transactions on Multimedia*, 27:6593–6605, 2025. 3

[49] 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 International Conference on Machine Learning*, volume 139, pages 8748–8763, 2021. 1, 3, 6, 7, 21, 22

[50] Frank Ruis, Gertjan Burghouts, and Doina Bucur. Independent prototype propagation for zero-shot compositionality. In *Advances in Neural Information Processing Systems*, volume 34, pages 10641–10653, 2021. 1, 3

[51] Nirat Saini, Khoi Pham, and Abhinav Shrivastava. Disentangling visual embeddings for attributes and objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13658–13667, 2022. 3, 7, 18, 22

[52] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35:14274–14289, 2022. 2, 3

[53] Elaine Sui, Xiaohan Wang, and Serena Yeung-Levy. Just shift it: Test-time prototype shifting for zero-shot generalization with vision-language models. In *Proceedings of IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 825–835. IEEE, 2025. 3, 18, 19, 21

[54] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: fully test-time adaptation by entropy minimization. In *Proceedings of the International Conference on Learning Representations*, 2021. 2, 3

[55] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022. 3

[56] Qingsheng Wang, Lingqiao Liu, Chenchen Jing, Hao Chen, Guoqiang Liang, Peng Wang, and Chunhua Shen. Learning conditional attributes for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11197–11206, 2023. 3, 18, 22

[57] Zehao Xiao, Shilin Yan, Jack Hong, Jiayin Cai, Xiaolong Jiang, Yao Hu, Jiayi Shen, Cheems Wang, and Cees GM Snoek. Dynaprompt: Dynamic test-time prompt tuning. In *Proceedings of the International Conference on Learning Representations*, 2025. 3

[58] Ling Xing, Hongyu Qu, Rui Yan, Xiangbo Shu, and Jinhui Tang. Locality-aware cross-modal correspondence learning for dense audio-visual events localization. *arXiv preprint arXiv:2409.07967*, 2024. 2

[59] Guangyue Xu, Joyce Chai, and Parisa Kordjamshidi. Gipcol: graph-injected soft prompting for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5774–5783, 2024. 3, 7, 22

[60] Xudong Yan, Songhe Feng, Yang Zhang, Jian Yang, Yueguan Lin, and Haojun Fei. Leveraging mllm embeddings and attribute smoothing for compositional zero-shot learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2179–2187, 2025. 1, 6, 17

[61] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014. 2, 6, 17, 18, 21

[62] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10899–10909, 2023. 3

[63] Ce Zhang, Simon Stepputtis, Katia Sycara, and Yaqi Xie. Dual prototype evolving for test-time generalization of vision-language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 32111–32136, 2024. 3, 5, 18, 19, 21

[64] Jingyi Zhang, Jiaxing Huang, Xiaoqin Zhang, Ling Shao, and Shijian Lu. Historical test-time prompt tuning for vision foundation models. In *Advances in Neural Information Processing Systems*, volume 37, pages 12872–12896, 2024. 3

[65] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *Proceedings of the European Conference on Computer Vision*, pages 493–510, 2022. 6

[66] Taolin Zhang, Jinpeng Wang, Hang Guo, Tao Dai, Bin Chen, and Shu-tao Xia. Boostadapter: Improving vision-language test-time adaptation via regional bootstrapping. In *Advances in Neural Information Processing Systems*, volume 37, pages 67795–67825, 2024. 3

[67] Tian Zhang, Kongming Liang, Ruoyi Du, Wei Chen, and Zhanyu Ma. Disentangling before composing: Learning invariant disentangled features for compositional zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(2):1132–1147, 2025. 18, 22

[68] Tian Zhang, Kongming Liang, Ruoyi Du, Xian Sun, Zhanyu Ma, and Jun Guo. Learning invariant visual representations for compositional zero-shot learning. In *Proceedings of the European Conference on Computer Vision*, pages 339–355, 2022. [2](#), [6](#), [7](#), [17](#), [21](#)

[69] Tian Zhang, Kongming Liang, Ke Zhang, and Zhanyu Ma. Learning conditional prompt for compositional zero-shot learning. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 1–6, 2024. [18](#), [22](#)

[70] Yang Zhang and Songhe Feng. Enhancing domain-invariant parts for generalized zero-shot learning. In *Proceedings of the ACM International Conference on Multimedia*, page 6283–6291, 2023. [6](#)

[71] Yang Zhang, Songhe Feng, and Jiazheng Yuan. Continual compositional zero-shot learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1724–1732, 2024. [3](#)

[72] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022. [3](#)

[73] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. [1](#), [3](#), [6](#), [7](#), [21](#), [22](#)

[74] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15659–15669, 2023. [3](#)

[75] Jinguo Zhu, Weiyun Wang, Zhe Chen, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. [18](#), [19](#)

Appendix

A Visual Adapter

Based on AdapterFormer [5], a set of learnable adapters is injected in parallel with the multi-head attention layer and feed-forward layer, respectively within each Transformer block of the CLIP visual encoder, while keeping the initial parameters frozen, as shown in Fig. 8. Given the input embedding e of each layer, the adapter is formulated as:

$$\text{Adapter}(e) = \mathbf{W}^{\text{up}}(\text{ReLU}(\mathbf{W}^{\text{down}}e)) , \quad (10)$$

where \mathbf{W}^{down} and \mathbf{W}^{up} are learnable linear layers employed for down-sampling and up-sampling, respectively. For each layer, the final output is obtained by adding the adapter output to the original layer output through a residual connection.

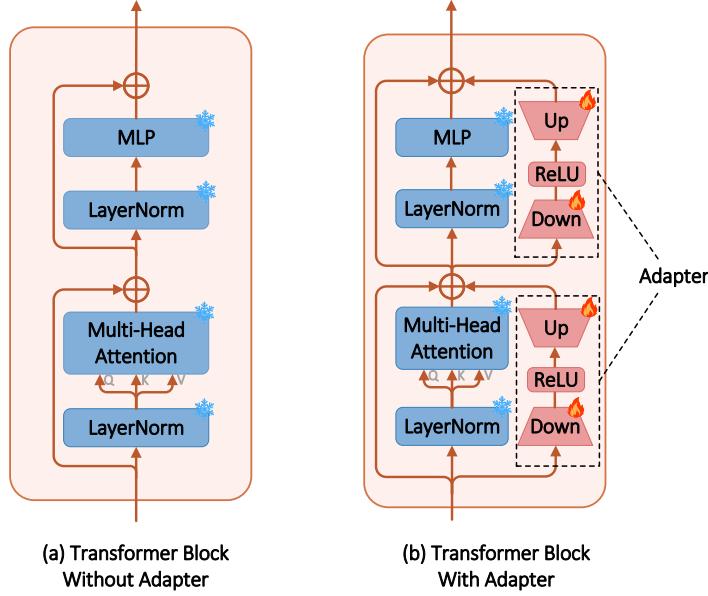


Figure 8: Transformer block within visual encoder of CLIP without and with Adapter.

B Theoretical Explanation of Prediction Entropy

In the absence of labeled target-composition samples, the Minimum Entropy Principle is adopted to improve model predictions. This principle is grounded in the cluster assumption of semi-supervised learning, which posits that the decision boundary should pass through low-density regions in the prediction space. By minimizing the entropy of the model predictions, we encourage high-confidence outputs and implicitly shift the decision boundary away from high-density data regions, thus reducing classification ambiguity.

With reference to [12], we formally define $x \sim \mathcal{D}_T$ as an unlabeled input, and define $p(y|x)$ as the predicted class distribution from the model. The prediction entropy of a single example is defined as:

$$\mathcal{H}(p(y|x)) = - \sum_{c=1}^{N_C} p(y = c|x) \log p(y = c|x) , \quad (11)$$

where N_C is the total number of classes. The entropy minimization objective is expressed as:

$$\mathcal{L}_{\text{entropy}} = \mathbb{E}_{x \sim \mathcal{D}_T} [\mathcal{H}(p(y|x))] . \quad (12)$$

Assuming the target data follows a continuous density distribution $q(x)$, the entropy loss becomes:

$$\mathcal{L}_{\text{entropy}} = \int \mathcal{H}(p(y|x))q(x) dx . \quad (13)$$

This formulation reveals that high-density regions (*i.e.*, large $q(x)$) contribute more to the loss. Therefore, minimizing this loss leads the model to produce low-entropy (*i.e.*, high-confidence) predictions in those regions, which implicitly encourages the decision boundary to lie in low-density areas of the prediction space.

On the other hand, to understand how entropy minimization sharpens model predictions, we analyze the gradient behavior in the prediction space. Let the model output logits $z = (z_1, \dots, z_{N_C})$, and the predicted probabilities be given by the softmax function:

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{N_C} e^{z_j}}. \quad (14)$$

Based on this, the gradient of entropy is computed with respect to the logits z_i using the chain rule. The derivative of $\mathcal{H}(p)$ with respect to p_i is:

$$\frac{\partial \mathcal{H}}{\partial p_i} = -\log p_i - 1, \quad (15)$$

and the Jacobian of the softmax function is:

$$\frac{\partial p_i}{\partial z_j} = \begin{cases} p_i(1 - p_i), & \text{if } i = j, \\ -p_i p_j, & \text{if } i \neq j. \end{cases} \quad (16)$$

Therefore, the gradient of entropy with respect to z_i becomes:

$$\frac{\partial \mathcal{H}}{\partial z_i} = \sum_{j=1}^{N_C} \frac{\partial \mathcal{H}}{\partial p_j} \cdot \frac{\partial p_j}{\partial z_i}. \quad (17)$$

This gradient encourages the logits to diverge, making one class score dominant over others, and thus increasing the confidence of the softmax prediction (*i.e.*, moving p closer to a one-hot vector). As a result, entropy minimization leads to sharper decision boundaries and effectively separates data clusters in the prediction space.

This microscopic gradient behavior aligns with the macroscopic Minimum Entropy Principle, both of which serve to place the decision boundary in low-density regions and enforce prediction consistency within high-density clusters.

C Dataset Introduction

We conduct our experiments on four CZSL datasets, *i.e.*, UT-Zappos [61], MIT-States [15], C-GQA [40], and Clothing16K [68]. UT-Zappos is a fine-grained fashion dataset consisting of different shoes (*e.g.*, heels, sandals) with their material (*e.g.*, leather, satin). MIT-States contains diverse real-world objects (*e.g.*, envelope, tower) and attributes (*e.g.*, folded, tight) collected by early image search engine technology. However, it suffers from considerable noise, with approximately 70% of the labels being incorrect [2, 68, 60]. C-GQA is a large-scale dataset composed of in-the-wild images with more general compositions (*e.g.*, marble countertop, striped pillow). For the low-noise dataset Clothing16K, the objects are mainly various kinds of clothing (*e.g.*, dress, shirt), while the attributes are the clothing colors (*e.g.*, pink, white). Table 7 reports the common data splits of the four datasets.

Table 7: Summary statistics of the four datasets used in our experiments.

Dataset	Composition			Train		Validation			Test		
	$ A $	$ O $	$ A \times O $	$ C^s $	$ \mathcal{X} $	$ C^s $	$ C^u $	$ \mathcal{X} $	$ C^s $	$ C^u $	$ \mathcal{X} $
UT-Zappos [61]	16	12	192	83	22998	15	15	3214	18	18	2914
MIT-States [15]	115	245	28175	1262	30338	300	300	10420	400	400	12995
C-GQA [40]	413	674	278362	5592	26920	1252	1040	7280	888	923	5098
Clothing16K [68]	9	8	72	18	7242	10	10	5515	9	8	3413

D Implementation Details

We use a single NVIDIA RTX 3090 GPU to train the base model during training and TOMCAT at test time using mixed-precision training [37] under the PyTorch framework [45]. The trainable prompt of CLIP ViT-L/14 in the training phase is initialized by "*a photo of*". The hyperparameters for the four datasets are listed in Table 8.

Table 8: Hyperparameter settings for UT-Zappos, MIT-States, C-GQA, and Clothing16K.

Hyperparameters	UT-Zappos	MIT-States	C-GQA	Clothing16K
The Base Model (Training Phase)				
Batch Size	128	64	16	128
Epochs	20	20	20	20
Prompt Dropout Rate	0.3	0.3	0	0.3
Adapter Downsampling Dimension	64	64	64	64
Adapter Dropout	0.1	0.1	0.1	0.1
Optimizer	Adam	Adam	Adam	Adam
Optimizer-Weight Decay	1e-5	1e-4	1e-5	1e-5
Optimizer-Learning Rate	5e-4	1e-4	1e-4	5e-4
Scheduler	StepLR	StepLR	StepLR	StepLR
Scheduler-Step Size	5	5	5	5
Scheduler-Gamma	0.5	0.5	0.5	0.5
TOMCAT (Test Phase)				
Batch Size	1	1	1	1
Image Number of Priority Queue	3	3	3	3
Optimizer	AdamW	AdamW	AdamW	AdamW
Optimizer-Epsilon	1e-3	1e-3	1e-3	1e-3
Optimizer-Weight Decay	1e-3	1e-4	1e-4	1e-3
Optimizer-Learning Rate	5e-6	1e-6	6.25e-6	5e-6
α	0	1.25	0.5	0.25
β	-	10	10	7.5
θ	1	1.5	2	1
λ	3.5	2.5	1.75	3.5

E Comparison with More Baselines

In this section, we compare our TOMCAT with three types of baselines on UT-Zappos, MIT-States, and C-GQA: (1) more CZSL methods; (2) online test-time adaptation (OTTA) of VLM methods; and (3) multimodal large language models (MLLMs).

Comparison with More CZSL Methods. TOMCAT is compared with more state-of-the-art CZSL methods on UT-Zappos [61], MIT-States [15], and C-GQA [40] in both closed-world and open-world settings in Table 17 and Table 18, respectively. Relative to Table 1, these additional baselines include LE+ [39], AttOp [41], TMN [46], SymNet [26], SCEN [25], OADis [51], ADE [13], CANET [56], CLPS [18], DBC [67], CoP [69], VisProd [20], KG-SP [21], and SAD-SP [33]. Our TOMCAT surpasses all other models and achieves state-of-the-art performance.

Comparison with OTTA of VLM Methods. Building on the base model obtained after training, we conduct experiments on the three datasets with OTTA of VLM approaches, including online-TPS [53], TDA [19], and DPE [63]. These methods also utilize unlabeled test data, making the comparison fairer, although the original comparison with CZSL methods is fair due to the absence of labels of test samples. In Table 9, TOMCAT yields 3.7%, 0.4%, and 0.4% AUC gains compared with OTTA of VLM methods on the three datasets, which demonstrates that TOMCAT also exhibits superior performance over OTTA methods on CZSL task under fair conditions.

Comparison with MLLMs. With the advancement of MLLMs, they can also be employed to generate image descriptions or categories. Therefore, we compare TOMCAT with two outstanding MLLMs (*i.e.*, LLaVA v1.5 [31] and InternVL-3 [75]) on UT-Zappos and MIT-States. Specifically, we provide the prompt “*Classify the image and output strictly two words in the form: ‘attribute object’ (an attribute must be generated). No other text. Example: ‘red apple’. Choose the attribute and object from their respective list. Attribute list: [Attribute List]. Object list: [Object List]*.” to

Table 9: Comparison with OTTA of VLM methods in the closed-world setting on UT-Zappos, MIT-States, and C-GQA.

Closed-world Methods	UT-Zappos				MIT-States				C-GQA			
	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen
online-TPS [53]	44.6	57.3	71.3	71.4	22.2	39.1	49.6	52.8	15.6	33.8	45.0	39.6
TDA [19]	41.6	54.6	69.0	70.6	22.0	38.7	49.5	52.9	14.4	32.0	44.7	37.2
DPE [63]	43.3	56.0	68.5	74.2	22.0	38.9	49.2	52.8	15.3	33.5	44.0	39.7
TOMCAT (Ours)	48.3	60.2	74.5	72.8	22.6	39.5	50.3	53.0	16.0	34.0	45.3	40.1

Table 10: Comparison with MLLMs on UT-Zappos and MIT-States.

Methods	UT-Zappos				MIT-States			
	Comp. \uparrow	Attr. \uparrow	Obj. \uparrow	Time (ms) \downarrow	Comp. \uparrow	Attr. \uparrow	Obj. \uparrow	Time (ms) \downarrow
LLaVA v1.5 [31]	0.32	10.17	2.13	463	5.72	1.06	21.77	857
InternVL-3 [75]	2.06	8.77	23.71	5770	4.43	8.53	48.28	728
TOMCAT (Ours)	44.88	58.54	77.45	47	30.92	41.00	56.58	145

MLLMs and obtain their predictions. Since MLLMs cannot produce class probabilities for computing AUC, we report the top-1 classification accuracy (%) for compositions (**Comp.**), attributes (**Attr.**), objects (**Obj.**), and the inference **Time** in Table 10. TOMCAT significantly outperforms them with less time overhead, indicating that MLLMs are primarily designed for text generation and are less effective for representation learning tasks (*e.g.*, recognition and retrieval). Moreover, the behavior of generative models is inherently difficult to control, making them unsuitable for recognition tasks.

F More Ablation Study

In this section, more ablation studies are presented in the closed-world setting.

Ablation study on fine-tuning CLIP with the training set. In Table 11, we respectively use the naive CLIP without fine-tuning on the training set and the base model fine-tuned on the training data as the starting points for testing. The latter significantly outperforms the former, indicating that while the naive CLIP exhibits strong zero-shot abilities, its general-purpose nature limits its effectiveness in the specific downstream task requiring composition reasoning under the zero-shot setting.

Influence of learning rate. Since our method simulates test-time improvement in real-world applications, the value of learning rate is crucial. According to Table 12, TOMCAT achieves optimal performance when the learning rate is set to 5e-6.

Table 11: Ablation study of fine-tuning CLIP on the training set of three datasets, respectively.

Dataset	tuning	AUC	HM	Seen	Unseen
UT-Zappos	✗	1.54	7.29	4.79	50.66
	✓	48.31	60.18	74.49	72.77
MIT-States	✗	11.42	26.79	30.88	46.50
	✓	22.55	39.45	50.32	52.95
C-GQA	✗	1.47	8.84	7.79	25.09
	✓	15.98	33.95	45.30	40.12

Table 12: Influence of learning rate (lr) of TOMCAT on UT-Zappos.

lr	AUC	HM	Seen	Unseen
1e-3	17.11	17.11	54.25	39.77
1e-4	28.66	46.14	70.19	45.21
1e-5	47.28	58.96	76.25	69.65
5e-6	48.31	60.18	74.49	72.77
1e-6	45.78	57.65	70.97	74.25
1e-7	43.69	55.77	68.82	74.35

Influence of the hyperparameter controlling multimodal balance α . In table 13, TOMCAT achieves the best performance on MIT-States when α is set to 1.25. An excessively large or small value of α leads the model to overemphasize one modality while neglecting the other, thereby undermining its discriminative capability.

Influence of the weighting coefficient of \mathcal{L}_{MCRL} λ . In TOMCAT, balancing the contributions of each loss term plays a crucial role. As shown in Table 14, the best performance on MIT-States is

achieved when $\lambda = 1.5$, indicating that an appropriate trade-off between the competing objectives leads to optimal model behavior.

Table 13: Influence of α on MIT-States.

α	AUC	HM	Seen	Unseen
0.25	22.24	39.27	49.58	52.86
0.5	22.20	39.22	49.50	52.81
0.75	22.34	39.47	49.75	52.89
1	22.36	39.38	49.92	52.83
1.25	22.55	39.45	50.32	52.95
1.5	22.43	39.31	49.96	52.95
1.75	22.41	39.31	49.79	53.04
2	22.42	39.17	49.79	53.12

Table 14: Influence of λ on MIT-States.

λ	AUC	HM	Seen	Unseen
0.25	22.50	39.46	50.08	52.99
0.5	22.53	39.41	50.13	52.93
0.75	22.50	39.40	50.04	52.92
1	22.51	39.47	50.04	52.91
1.25	22.48	39.42	50.00	52.93
1.5	22.55	39.45	50.32	52.95
1.75	22.53	39.48	50.17	52.91
2	22.46	39.45	50.01	52.92

Ablation Study of Prediction Probability in Eq. 6. In Table 15, we observe that relying solely on visual prototypes yields poor performance, likely due to the large intra-class variance in images, which limits the discriminative power of the model. Incorporating both textual and visual prototypes better exploits the complementary multimodal strengths, enabling TOMCAT to achieve good performance.

Table 15: Ablation Study of prediction probability on three datasets. Text- and visual-only denote that the prediction probability is derived from only textual prototypes and visual prototypes, respectively.

Prediction Probability	UT-Zappos				MIT-States				C-GQA			
	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen
Text-only	48.31	60.18	74.49	72.77	22.1	38.95	49.33	52.8	15.79	33.76	45.03	39.86
Visual-only	1.66	0	67.45	4.92	15.66	33.76	16.76	41.48	3.95	16.65	29.29	14.95
Multimodal (TOMCAT)	48.31	60.18	74.49	72.77	22.55	39.45	50.32	52.95	15.98	33.95	45.30	40.12

G More Qualitative Analysis

In this section, we report more qualitative analysis of TOMCAT in the closed-world setting.

Time and memory analysis. As shown in Table 16, we present the time, latency, and memory occupation of the base model and TOMCAT. Our TOMCAT achieves substantial performance improvement with minimal additional computational resources at test time.

Table 16: Comparison of time, latency, and memory between TOMCAT and the base model (w/o TOMCAT) on UT-Zappos and MIT-States. Time means that average testing time across all samples. Latency indicates time from input to prediction output for per sample (excluding backpropagation). Memory represents the GPU memory occupied by the model during testing.

Dataset	Time		Latency		Memory	
	TOMCAT	w/o TOMCAT	TOMCAT	w/o TOMCAT	TOMCAT	w/o TOMCAT
UT-Zappos	47ms	36ms	35ms	32ms	4044MB	3801MB
MIT-States	145ms	70ms	112ms	37ms	4180MB	4175MB

More case study. In Fig. 9 and Fig. 10, we present more case studies on UT-Zappos and MIT-States, respectively. As shown, by leveraging our designed algorithm with unlabeled data at test time, TOMCAT significantly improves the accuracy of composition predictions compared to the base model. In some failure cases (*e.g.*, the image labeled as `new bus` in MIT-States), our method is also capable of making correct predictions, though it focuses on aspects different from the labels.

H Broader Impacts

In this work, we recognize that the unseen compositions resulting from the recombination of attributes and objects lead to the distribution shift of label space. Therefore, we aim to establish a more reliable



Figure 9: More case study on UT-Zappos. We compare TOMCAT (Ours) with the base model (BM) after training. The successful and failure results are marked in green and red, respectively.



Figure 10: More case study on MIT-States.

compositional reasoning system by leveraging test samples to enhance composition modeling, which is more in line with the application and development in real-world scenarios. To the best of our knowledge, our method has no potential negative societal impact.

I Licenses

We have explicitly cited the datasets and models used in our paper. Their licenses are listed as follows.

Datasets. UT-Zappos [61] and MIT-States [15] are two early-proposed datasets commonly used in CZSL. Their creators and owners have not declared any license and have allowed non-commercial research use. C-GQA [40] is under CC BY 4.0 license. Clothing16K [68] is under CC0 license.

Models. In this work, our TOMCAT is built on existing code repositories: CLIP [49], CoOp [73], AdapterFormer [5], Troika [14], TDA [19], TPS [53], and DPE [63]. The code implementations used in our paper are all under MIT license.

Table 17: Comparison with more baselines in the closed-world setting on UT-Zappos, MIT-States, and C-GQA. The best results are displayed in **boldface**, and the second-best results are underlined.

Closed-world Methods	UT-Zappos				MIT-States				C-GQA			
	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen
LE+ [39] (<i>CVPR'17</i>)	25.7	41.0	53.0	61.9	2.0	10.7	15.0	20.1	0.6	5.3	16.1	5.0
AttOp [41] (<i>CVPR'18</i>)	25.9	40.8	59.8	54.2	1.6	9.9	14.3	17.4	0.3	2.9	11.8	3.9
TMN [46] (<i>ICCV'19</i>)	29.3	45.0	58.7	60.0	2.9	13.0	20.2	20.1	1.1	7.7	21.6	6.3
SymNet [26] (<i>CVPR'20</i>)	23.9	39.2	53.3	57.9	3.0	16.1	24.4	25.2	1.8	9.8	25.2	9.2
SCEN [25] (<i>CVPR'22</i>)	32.0	47.8	63.5	63.1	5.3	18.4	29.9	25.2	2.9	12.4	28.9	12.1
OADis [51] (<i>CVPR'22</i>)	30.0	44.4	59.5	65.5	5.9	18.9	31.1	25.6	-	-	-	-
ADE [13] (<i>CVPR'23</i>)	35.1	51.1	63.0	64.3	-	-	-	-	5.2	18.0	35.0	17.7
CANET [56] (<i>CVPR'23</i>)	33.1	47.3	61.0	66.3	5.4	17.9	29.0	26.2	3.3	14.5	30.0	13.2
CLPS [18] (<i>TMM'25</i>)	37.2	51.8	63.2	70.1	-	-	-	-	5.6	18.9	35.1	19.0
DBC [67] (<i>TPAMI'25</i>)	35.4	51.3	63.0	67.5	-	-	-	-	5.0	18.0	35.2	17.8
CLIP [49] (<i>ICML'21</i>)	5.0	15.6	15.8	49.1	11.0	26.1	30.2	46.0	1.4	8.6	7.5	25.0
CoOp [73] (<i>ICCV'22</i>)	18.8	34.6	52.1	49.3	13.5	29.8	34.4	47.6	4.4	17.1	20.5	26.8
Co-CGE [36] (<i>TPAMI'22</i>)	36.3	49.7	63.4	71.3	17.0	33.1	46.7	45.9	5.7	18.9	34.1	21.2
CSP [42] (<i>ICLR'23</i>)	33.0	46.6	64.2	66.2	19.4	36.3	46.6	49.9	6.2	20.5	28.8	26.8
DFSP(i2t) [34] (<i>CVPR'23</i>)	32.1	45.1	64.2	66.4	20.7	37.2	47.4	52.4	8.7	24.3	35.6	29.3
DFSP(BiF) [34] (<i>CVPR'23</i>)	33.5	47.1	63.3	69.2	20.8	37.7	47.1	52.8	9.9	26.2	36.5	32.0
DFSP(2t) [34] (<i>CVPR'23</i>)	36.0	47.2	66.7	71.7	20.6	37.3	46.9	52.0	10.5	27.1	38.2	32.0
GIPCOL [59] (<i>WACV'24</i>)	36.2	48.8	65.0	68.5	19.9	36.6	48.5	49.6	7.1	22.5	31.9	28.4
Troika [14] (<i>CVPR'24</i>)	41.7	54.6	66.8	73.8	22.1	39.3	49.0	53.0	12.4	29.4	41.0	35.7
CDS-CZSL [28] (<i>CVPR'24</i>)	39.5	52.7	63.9	74.8	22.4	39.2	50.3	52.9	11.1	28.1	38.3	34.2
CoP [69] (<i>ICME'24</i>)	36.2	51.2	64.8	67.3	19.7	36.4	47.0	50.9	7.3	21.7	33.3	27.0
PLID [3] (<i>ECCV'24</i>)	38.7	52.4	67.3	68.8	22.1	39.0	49.7	52.4	11.0	27.9	38.8	33.0
IMAX [17] (<i>TPAMI'25</i>)	40.6	54.2	69.3	70.7	21.9	39.1	48.7	53.8	12.8	29.8	39.7	35.8
ClusPro [47] (<i>ICLR'25</i>)	46.6	58.5	70.7	76.0	<u>23.8</u>	<u>40.7</u>	52.1	54.0	14.9	<u>32.8</u>	44.3	37.8
TOMCAT (Ours)	48.3	60.2	74.5	72.8	<u>22.6</u>	<u>39.5</u>	<u>50.3</u>	53.0	16.0	34.0	<u>45.3</u>	40.1

Table 18: Comparison with more baselines in the open-world setting on UT-Zappos, MIT-States, and C-GQA. We report the results of CDS-CZSL* [28] using the same post-training feasibility calibration [35] as our TOMCAT and other baselines use.

Open-world Methods	UT-Zappos				MIT-States				C-GQA			
	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen
LE+ [39] (<i>CVPR'17</i>)	16.3	30.5	60.4	36.5	0.3	2.7	14.2	2.5	0.1	1.0	19.2	0.7
AttOp [41] (<i>CVPR'18</i>)	13.7	29.4	50.9	34.2	0.7	4.7	16.6	5.7	-	-	-	-
TMN [46] (<i>ICCV'19</i>)	8.4	21.7	55.9	18.1	0.1	1.2	12.6	0.9	-	-	-	-
SymNet [26] (<i>CVPR'20</i>)	18.5	34.5	53.3	44.6	0.8	5.8	21.4	7.0	0.4	3.3	26.7	2.2
VisProd [20] (<i>NeurIPS'21</i>)	19.7	36.9	54.6	42.8	0.7	5.6	20.9	5.8	0.3	2.8	24.8	1.7
KG-SP [21] (<i>CVPR'22</i>)	26.5	42.3	61.8	52.1	1.3	7.4	28.4	7.5	0.8	4.7	31.5	2.9
ADE [13] (<i>CVPR'23</i>)	27.1	44.8	62.4	50.7	-	-	-	-	1.4	7.6	35.1	4.8
SAD-SP [33] (<i>TPAMI'24</i>)	28.4	44.0	63.1	54.7	1.4	7.8	29.1	7.6	1.0	5.9	31.0	3.9
DBC [67] (<i>TPAMI'25</i>)	28.6	44.9	63.0	52.8	-	-	-	-	1.4	7.6	35.6	4.7
CLIP [49] (<i>ICML'21</i>)	2.2	11.2	15.7	20.6	3.0	12.8	30.1	14.3	0.3	4.0	7.5	4.6
CoOp [73] (<i>ICCV'22</i>)	13.2	28.9	52.1	31.5	2.8	12.3	34.6	9.3	0.7	5.5	21.0	4.6
Co-CGE [36] (<i>TPAMI'22</i>)	28.4	45.3	59.9	56.2	5.6	17.7	38.1	20.0	0.9	5.3	33.2	3.9
CSP [42] (<i>ICLR'23</i>)	22.7	38.9	64.1	44.1	5.7	17.4	46.3	15.7	1.2	6.9	28.7	5.2
DFSP(i2t) [34] (<i>CVPR'23</i>)	26.4	41.2	64.3	53.8	6.7	19.1	47.2	18.2	2.0	9.0	35.6	6.5
DFSP(BiF) [34] (<i>CVPR'23</i>)	27.6	42.7	63.5	57.2	6.7	19.2	47.1	18.1	2.4	10.6	36.4	7.6
DFSP(2t) [34] (<i>CVPR'23</i>)	30.3	44.0	66.8	60.0	6.8	19.3	47.5	18.5	2.4	10.4	38.3	7.2
GIPCOL [59] (<i>WACV'24</i>)	23.5	40.1	65.0	45.0	6.3	17.9	48.5	16.0	1.3	7.3	31.6	5.5
Troika [14] (<i>CVPR'24</i>)	33.0	47.8	66.4	61.2	7.2	20.1	48.8	18.7	2.7	10.9	40.8	7.9
CDS-CZSL* [28] (<i>CVPR'24</i>)	32.1	48.0	64.7	60.4	-	-	-	-	2.6	10.9	38.2	8.0
PLID [3] (<i>ECCV'24</i>)	30.8	46.6	67.6	55.5	7.3	20.4	49.1	18.7	2.5	10.6	39.1	7.5
IMAX [17] (<i>TPAMI'25</i>)	32.3	47.5	68.4	57.3	7.6	21.4	<u>50.2</u>	18.6	2.6	11.2	38.7	7.9
ClusPro [47] (<i>ICLR'25</i>)	39.5	54.1	71.0	66.2	<u>9.3</u>	23.0	51.2	22.1	3.0	<u>11.6</u>	41.6	8.3
TOMCAT (Ours)	43.7	57.9	74.1	<u>65.8</u>	<u>8.2</u>	<u>21.7</u>	49.2	<u>21.0</u>	4.2	14.2	45.1	10.6

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: We have clearly stated our contributions and scope of the paper in Abstract and Introduction (Sec. 1).

Guidelines:

- The answer NA 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 NA 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: We have discussed the limitations of our work in Conclusion (Sec. 5).

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not provide any theoretical results.

Guidelines:

- The answer NA 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: We provide the model framework and implementation in Method (Sec. 3), Experiment Setup (Sec. 4.1), and Implementation Details (Appendix D). Therefore, we believe that the readers can replicate the proposed TOMCAT by following our instructions.

Guidelines:

- The answer NA 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: All datasets are publicly available through the links provided in the respective papers where they are proposed. The source code of our method will be released at <https://github.com/xud-yan/TOMCAT>.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We report all the training and test details in our paper, including data splits (Appendix C), analysis of hyper-parameters (Sec. 4.3), and the implementation details (Appendix D).

Guidelines:

- The answer NA 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: We vary the test order using different random seeds to conduct statistical significance tests in Ablation Study (Appendix 4.3).

Guidelines:

- The answer NA 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: We present the time consumption and GPU memory usage of our method in More Qualitative Analysis (Appendix [G](#)).

Guidelines:

- The answer NA 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: We have reviewed the NeurIPS Code of Ethics and make sure that our research conform with it in every respect.

Guidelines:

- The answer NA 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: We have discussed both potential positive and negative societal impacts of our work in Broader Impacts (Appendix [H](#)).

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We are convinced that the paper dose not pose any risk.

Guidelines:

- The answer NA 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: We have explicitly cited all the assets we used in our paper. Additionally, the licenses of our used datasets and code are presented in Licenses section (Appendix I).

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not introduce any new assets.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Our proposed method does not involve LLM.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.