

Annotation-Free Class-Incremental Learning

Hari Chandana Kuchibhotla^{‡*}, K S Ananth^{‡*}, Vineeth N Balasubramanian^{‡◊}

[‡] Indian Institute of Technology Hyderabad, India [◊] Microsoft Research India

{ai20resch11006, cs22btech11029, vineethnb}@iith.ac.in, vineeth.nb@microsoft.com

Abstract

Despite significant progress in continual learning ranging from architectural novelty to clever strategies for mitigating catastrophic forgetting; most existing methods rest on a strong but unrealistic assumption: “the availability of labeled data throughout the learning process”. In real-world scenarios, however, data often arrives sequentially and without annotations, rendering conventional approaches impractical. In this work, we revisit the fundamental assumptions of continual learning and ask: Can current systems adapt when labels are absent and tasks emerge incrementally over time? To this end, we introduce Annotation-Free Class-Incremental Learning (AF-CIL), a more realistic and challenging paradigm where unlabeled data arrives continuously, and the learner must incrementally acquire new classes without any supervision. To enable effective learning under AF-CIL, we propose CrossWorld-CL, a Cross Domain World Guided Continual Learning framework that incorporates external world knowledge as a stable auxiliary source. The method retrieves semantically related ImageNet classes for each downstream category, maps downstream and ImageNet features through a cross-domain alignment strategy and finally introduce a novel replay strategy. This design lets the model uncover semantic structure without annotations while keeping earlier knowledge intact. Across four datasets, CrossWorld-CL surpasses CLIP baselines and existing continual and unlabeled learning methods, underscoring the benefit of world knowledge for annotation-free continual learning.

1. Introduction

Continual Learning [9, 20, 22, 23] has evolved as a powerful paradigm to enable models to learn from a stream of tasks without retraining from scratch. Over the years, research in CL has largely revolved around mitigating catastrophic forgetting, the tendency of neural networks to

overwrite previously learned knowledge when exposed to new data. While architectural innovations [23], rehearsal buffers [9], and regularization strategies [22] have led to notable progress, the majority of these methods are built upon a strong but often unrealistic assumption: *that labeled data is always available for every incoming task*. In reality, continual data annotation is expensive, time-consuming, and cognitively demanding, placing an unsustainable burden on human annotators. Moreover, handling unlabeled data is inherently challenging due to distributional shifts between sequentially arriving data streams and test distributions. Maintaining accuracies across tasks with such partial information where only training images and labels are available but are not paired is a challenge by itself. For example, online retail platforms continually receive new product images from vendors, many of which arrive without labels but still need to be identified and assigned to the correct category. As this process repeats, both the volume of data and the number of categories grow over time. In addition to this, with the rapidly growing privacy concerns, exemplar-based methods violate this condition. Existing methods fail miserably at addressing these challenges. To address these issues, we introduce the much needed-**Annotation-Free Class Incremental learning** paradigm that can (i) learn effectively from unlabeled, sequentially arriving data, (ii) maintain high inference accuracy on both old and new classes, and (iii) prevent forgetting without relying on stored exemplars.

Traditional continual unsupervised representation learning frameworks aim to preserve the quality of learned features as unlabeled data arrives over time. These approaches often rely on contrastive or clustering-based self supervision methods such as SimCLR [3], BYOL [6], MoCo [8], and DINO [2], combined with strategies like momentum updates or replay buffers to reduce forgetting. However, they work entirely within visual feature spaces, usually using ResNet [7] backbones, and therefore struggle to adapt when the underlying semantic structure of tasks changes. Their representations remain limited because they do not incorporate language or broader world knowledge.

*equal contribution

To address these challenges, we propose *CrossWorld CL*, a Cross Domain World Guided Continual Learning framework that uses external world knowledge as an auxiliary supervision. We rely on the broad visual corpora that can encode general semantic structure, which can guide learning when annotations are missing. A proxy equivalent of world knowledge that can be publicly accessible is the widely used ImageNet[5] dataset due to its rich visual variability and well formed category boundaries that is otherwise missing in the unlabeled stream.

Our proposed approach exploits the latent semantic structure between the downstream unlabeled classes and the ImageNet classes to retrieve semantically aligned auxiliary data under Task-aware World Knowledge Distillation stage. This retrieved data acts as proxy supervision that guides the model through a cross domain loss, helping it bridge the gap between unlabeled and labeled domains while also avoiding privacy concerns. While dealing with the downstream data, motivated by prior work [10], we make use of GPT3 [1] generated text descriptions for all downstream classes. Since averaged text embeddings are known to capture semantics more effectively than individual descriptions, we employ CLIP [16] to obtain averaged text embeddings, which serve as the initial textual features under Semantic Expansion through LLM stage. These embeddings are then trained to align with the visual features within our framework. To enhance the visual supervision, we employ DINO [2], which is known for its strong visual backbone, to generate pseudo labels that guide the CLIP [16] model during training. A mapping mechanism using both downstream data and the retrieved ImageNet data is designed between the DINO and CLIP feature spaces to enable smooth knowledge transfer under Dual-Supervised Visual-Semantic Alignment stage. Once DINO [2] is trained to predict reliable pseudo labels, we integrate external world knowledge into training a prompts infused CLIP visual model using cross-domain alignment losses which greatly aids in learning the hidden semantics across domains and improves the label predictions under Prompt-guided Cross-domain Alignment stage. Lastly we use the world knowledge as pseudo-exemplars under Replay Strategy stage. By grounding continual learning in publicly available world knowledge instead of private exemplars, our framework achieves strong generalization and mitigates catastrophic forgetting without any risk of data leakage.

Our main contributions are as follows:

- We introduce *Annotation-Free Class Incremental Learning* (AF-CIL), a new paradigm for unlabeled sequential data.
- We introduce CrossWorld CL, a Cross Domain World Guided Continual Learning framework, that integrates external world knowledge and introduce a privacy pre-

serving replay strategy using semantically related samples instead of storing task-specific exemplars to address Annotation-Free CIL.

- We extend the CLIP–DINO synergy with cross domain alignment losses for better pseudo labeling and semantic alignment.
- We show strong performance gains across four datasets, addressing a key gap in continual learning research.

2. Related Works

Zero-Shot Recognition. Zero-shot CLIP [16] serves as a strong starting point for recognition without labels, and when adapted continually emerges into the Continual-CLIP [19] framework. However, performance drops as the number of classes increases, since fixed text prompts struggle to separate fine grained categories and class overlap grows over time. Some level of adaptation or retraining becomes necessary to maintain stability making it not suitable for our task as it does not scale well with expanding class space and fails to maintain performance without labeled supervision.

Unlabeled image recognition. LaFTer [13] improves zero shot classification by aligning CLIP’s text embeddings with visual clusters from unlabeled data using a contrastive loss, effectively tuning the classifier without any labels. NoLA [10] extends this idea by combining CLIP and DINO, where DINO provides pseudo labels and CLIP’s text features serve as semantic anchors, refined through a cross space alignment loss. Both methods enhance zero shot generalization on static unlabeled datasets, but they operate in a single adaptation stage and assume fixed classes. Their lack of mechanisms for continual updates, privacy preservation, and long term knowledge retention makes them unsuitable for our sequential unlabeled learning setting.

Continual Unsupervised Representation Learning. Approaches such as SimCLR [3], BYOL [6], MoCo [8], and DINO [2] maintain representations over time using self supervised objectives, but they operate solely within the visual space, typically with ResNet [7] based backbones. Their representations lack the flexibility and semantic richness of vision language models, making them unsuitable for capturing evolving category relationships. While they stabilize visual features, they cannot achieve meaningful class level recognition or adapt to semantic drift across tasks. In contrast, our setting demands models that align visual and textual semantics to remain consistent as new classes emerge.

Continual Learning. L2P [20], DualPrompt [21], CODAPrompt [18], MoEAdapter [23], RAPF [9], and GIFT [22] represent recent advances in continual learning that mitigate forgetting through prompt tuning, adapter modules, or routing strategies. L2P [20] learns task specific prompts, DualPrompt [21] separates general and task dependent prompts, and CODAPrompt [18] organizes

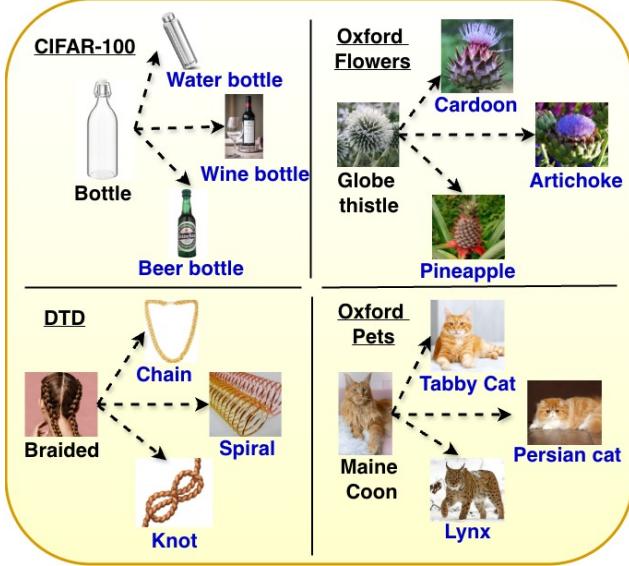


Figure 1. Examples of retrieved ImageNet classes for four downstream dataset categories; Bottle, Globe Thistle, Braided, and Maine Coon. The retrieved neighbors show clear semantic similarity to each target class, illustrating how our method gathers meaningful auxiliary supervision from ImageNet.

prompts in a hierarchical manner. MoEAdapter [23] employs expert adapters for task routing, while RAPF [9] and GIFT [22] use dynamic parameter allocation and synthetic feature replay to preserve knowledge. Although these methods achieve strong results in labeled settings, they are all heavily dependent on annotated data to guide prompt or adapter optimization and to define task boundaries. Without labeled supervision or stored exemplars, their mechanisms for retaining and transferring knowledge collapse, making them unsuitable for annotation-free continual learning.

Test-time tuning approaches. TPT [17] adapts model during inference using entropy or consistency objectives. They handle small domain shifts but lack long term memory and semantic expansion. TDA [11] proposes a lightweight framework to adapt vision-language models like CLIP during inference using entropy minimization and feature alignment, without full retraining. It achieves fast and efficient adaptation to distribution shifts, but it operates only at test time and lacks mechanisms for long term continual learning or handling sequential unlabeled data. While test time training methods adapt quickly to shifts, they lack long term learning ability and cannot handle sequential unlabeled tasks.

3. Methodology

Preliminaries. We begin by defining the key notations and components used in our framework, illustrated in Figure 2. Let the unlabeled downstream data at task t be denoted as

$\{x_{D_i}^t\}_{i=1}^{N_D}$, representing a set of images sampled from a distribution \mathcal{X}_D^t , with no labels available, where N_D denotes the number of samples during task t . The corresponding ground truth labels, which remain unknown during training, are denoted as $\{y_{D_i}^t\}_{i=1}^{N_D}$, where $y_{D_i}^t \in Y_D^t$. We further consider ImageNet dataset as our proxy for world knowledge, represented as $\{(x_{I_i}^t, y_{I_i}^t)\}_{i=1}^{N_I}$, where each $x_{I_i}^t \in \mathcal{X}_I$ is an ImageNet image and $y_{I_i}^t \in \mathcal{Y}_I$ is its corresponding semantic category label, where N_I denotes the number of samples of ImageNet dataset. Subscript D denotes the data is from downstream dataset and subscript I denotes the data is from ImageNet dataset. At test time, the model is evaluated on the cumulative test set containing samples from all tasks seen so far, defined as $\mathcal{T}_{\text{test}}^t = \bigcup_{k=1}^t \{(x_{\text{test}}^k, y_{\text{test}}^k)\}$ which spans the entire label space that the model has seen so far. Our framework builds upon three primary feature extractors: the CLIP text encoder f_t , the CLIP image encoder f_I , and the DINO image encoder f_D . The objective is to leverage the semantically rich supervision from ImageNet through both its images $x_{I_i}^t$ and labels $y_{I_i}^t$ to generate reliable and accurate pseudo labels for the unlabeled downstream samples $x_{D_i}^t$. This enables effective adaptation of the model to new classes in a continual and annotation-free manner.

Stage-1: Task-Aware World Knowledge Distillation.

The goal of this step is to distill task-relevant knowledge from large-scale external data such as ImageNet and use it as auxiliary supervision for the current downstream task. Given the set of downstream class names Y_D^t and the set of ImageNet class names \mathcal{Y}_I , we first encode all class names into the CLIP text embedding space using the text encoder f_t . We then compute the cosine similarity between each downstream class name and all ImageNet class names to identify the most semantically related ones:

$$\text{Sim}(c, c') = \frac{f_t(c) \cdot f_t(c')}{\|f_t(c)\| \|f_t(c')\|}, \quad c \in Y_D^t, \quad c' \in \mathcal{Y}_I.$$

For every downstream class c , we select its top- K most similar ImageNet classes based on this similarity measure. The corresponding ImageNet samples and their labels $(x_{I_i}^t, y_{I_i}^t)$ belonging to these selected categories are then aggregated to construct an auxiliary supervised dataset:

$$\mathcal{A}^t = \bigcup_{c \in Y_D^t} \{(x_{I_i}^t, y_{I_i}^t) \mid y_{I_i}^t \in \text{Top-}K(c)\}.$$

This auxiliary dataset captures only the portion of world knowledge that is semantically relevant to the downstream task. The distilled ImageNet classes corresponding to each downstream class from four different datasets are illustrated in Figure 1. It can be observed that the retrieved ImageNet categories are highly coherent and semantically relevant to their respective downstream classes. For instance, the

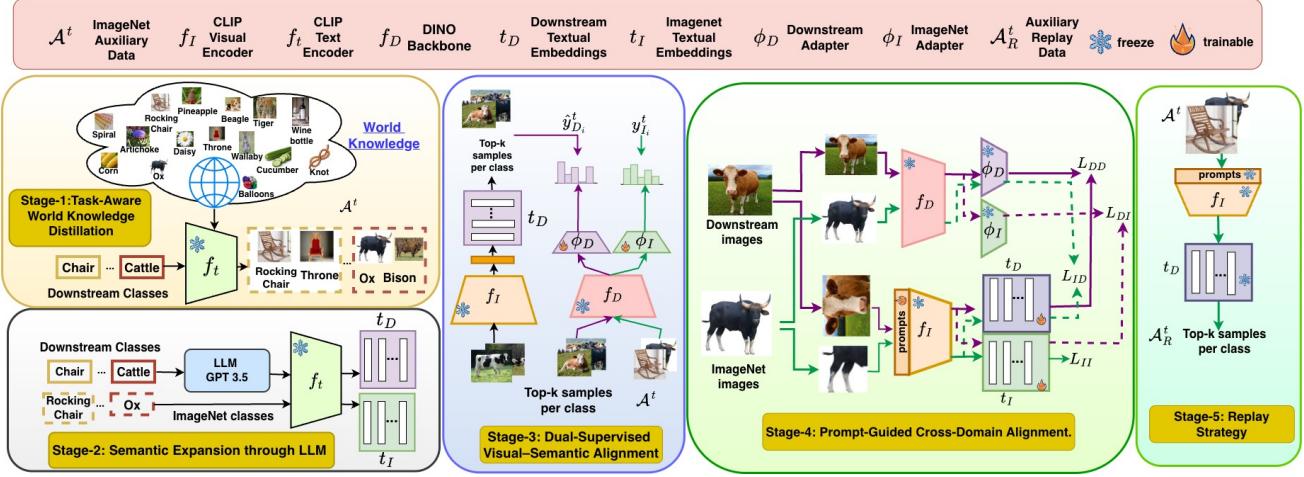


Figure 2. An overview of our annotation-free continual learning framework, which integrates world knowledge, multimodal alignment for better label prediction, and replay from public ImageNet data to learn new classes over time without the need to store downstream samples. **Purple** indicates downstream data flow and **Green** indicates ImageNet data flow. Stage 1: Task-aware world knowledge distillation retrieves semantically related ImageNet categories for each downstream class, forming an auxiliary supervision pool \mathcal{A}^t . Stage 2: An LLM expands downstream labels into diverse descriptions, which are encoded with CLIP to obtain robust text prototypes for both downstream (t_D) and ImageNet (t_I) sets. Stage-3: Dual-supervised visual–semantic alignment maps DINO features into CLIP space using pseudo-labeled downstream samples and supervised ImageNet pairs. Stage-4: Prompt-guided cross-domain alignment tunes the CLIP image encoder and text prototypes using multiple alignment losses (L_{DD} , L_{DI} , L_{ID} , L_{II}) across downstream and ImageNet branches. Stage-5: A replay strategy constructs a lightweight exemplar set from aligned ImageNet images, enabling rehearsal without storing private downstream data \mathcal{A}_R^t .

downstream class "bottle" retrieves ImageNet categories such as "water bottle", "beer bottle", and "wine bottle". These examples demonstrate that the proposed semantic retrieval effectively captures both visual and conceptual similarities across domains.

Stage-2: Semantic Expansion through LLM. In this stage, we enhance the textual representations of downstream classes by enriching them with external world knowledge. Given the label set Y_D^t for task t , we first generate descriptive text prompts for each class using a large language model (LLM) such as GPT-3 [1] following [10]. These descriptions provide diverse semantic cues about the visual category, capturing appearance, context, and relationships with other concepts. Each textual description is then encoded using the CLIP text encoder f_t . For every class $c \in Y_D^t$, we sample M textual descriptions and obtain their embeddings through f_t , followed by averaging to form a single representative text prototype: $\mathbf{t}_D = \frac{1}{M} \sum_{m=1}^M f_t(p_c^m)$, where p_c^m denotes the m^{th} textual prompt of class c . This averaging step yields a robust and semantically rich text feature that reduces bias from individual descriptions. For the auxiliary ImageNet dataset, we compute textual embeddings for each class label $y_{I_i}^t \in \mathcal{A}^t$ using the same CLIP text encoder f_t , and denote them as t_I . The embeddings of the downstream and ImageNet classes thus lie within

a shared semantic space, which facilitates effective cross-domain alignment in the later stages.

Stage-3: Dual-Supervised Visual–Semantic Alignment. We aim to align the DINO visual space with the CLIP semantic space by jointly utilizing the pseudo-labeled downstream data and the auxiliary ImageNet dataset. We employ DINO as the visual backbone due to its strong self-supervised learning ability and its proven effectiveness in capturing rich object-centric features without requiring labels. Unlike CLIP [16], which focuses on global image–text alignment, DINO [2] learns discriminative visual representations that generalize well across unseen domains. For the downstream data samples $x_{D_i}^t \in X_D^t$, we first obtain pseudo labels using the CLIP image encoder f_I and the downstream text prototypes t_D . The predicted label for each sample is computed as

$$\hat{y}_{D_i}^t = \arg \max_{c \in Y_D^t} \text{Sim}(f_I(x_{D_i}^t), \mathbf{t}_D^c).$$

where $\text{Sim}(\cdot)$ denotes the cosine similarity between the image and text embeddings. To ensure label reliability, we select the top- K samples with the highest prediction confidence for each class and keep them aside for further supervision. Next, our goal is to use both the pseudo-labeled downstream samples $\{(x_{D_i}^t, \hat{y}_{D_i}^t)\}$ and the auxiliary supervised ImageNet data \mathcal{A}^t , to learn a mapping between the DINO and CLIP spaces. To achieve this, we

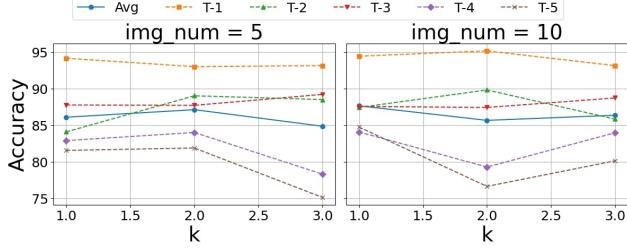


Figure 3. Effect of varying the number of retrieved ImageNet classes (k) and images per class ($\text{img_num} = 5, 10$) on task-wise accuracy across all five tasks for Oxford Pets dataset. Increasing img_num generally stabilizes performance, especially for later tasks Task-4 and Task-5.

freeze the DINO backbone f_D and attach two lightweight adapter networks; one for the downstream data and one for the ImageNet data denoted as ϕ_D and ϕ_I , respectively. Each adapter outputs logits corresponding to the number of classes in its domain, that is, $|\phi_D| = |\mathcal{Y}_D^t|$ and $|\phi_I| = |\mathcal{Y}_I^t|$. The downstream adapter ϕ_D is trained using the pseudo labels $\hat{y}_{D_i}^t$ as supervisory signals, while the ImageNet adapter ϕ_I is trained using the available ground-truth labels $y_{I_i}^t$. The learning objective minimizes the cross-entropy loss across both domains:

$$\mathcal{L}_{\text{map}} = \mathcal{L}_{\text{CE}}(\phi_D(f_D(x_{D_i}^t)), \hat{y}_{D_i}^t) + \mathcal{L}_{\text{CE}}(\phi_I(f_D(x_{I_i}^t)), y_{I_i}^t).$$

This step allows the DINO representations to be semantically grounded in the CLIP space by leveraging both pseudo and real supervision signals. As a result, the downstream DINO features become aligned with the language-driven CLIP embeddings, enabling better generalization and improved performance in the annotation-free continual learning setting.

Stage 4: Prompt-Guided Cross-Domain Alignment. In this stage, we perform cross-domain feature alignment to unify the representations of the downstream and auxiliary ImageNet datasets within a shared semantic space. Unlike the previous stage, where DINO was aligned with CLIP through adapter-based mapping, this stage focuses on refining the CLIP space itself to better capture task-specific semantics. To achieve this, we introduce learnable visual prompts into the CLIP image encoder, resulting in a prompted encoder denoted as f_{I_p} . These prompts are inserted into the patch token sequence of the vision transformer and act as lightweight, learnable context tokens that allow the encoder to adapt its visual representations to the downstream task while retaining CLIP’s pretrained generalization ability. The trainable parameters in this stage are the visual prompts within f_{I_p} and the textual embeddings corresponding to the downstream and ImageNet classes, namely t_D and t_I . Together, these parameters enable fine-grained semantic adaptation across domains under minimal

supervision. In this stage, the DINO backbone f_D and both adapters ϕ_D and ϕ_I are frozen. They now act as reliable pseudo-label generators to guide the learning of the prompted CLIP encoder. Every downstream image $x_{D_i}^t$ and auxiliary ImageNet image $x_{I_i}^t$ undergo two augmentations; an *identity transformation* and a *strong augmentation*. The identity-augmented samples are passed through the DINO encoder f_D to generate pseudo labels, while the strongly augmented samples are passed through the prompted CLIP image encoder f_{I_p} . The key contribution of our work lies in the proposed cross-domain alignment mechanism, where the DINO backbone features are passed through both the downstream adapter ϕ_D and the ImageNet adapter ϕ_I to generate same domain and cross-domain pseudo labels, while the prompted CLIP visual features are projected onto the corresponding text embeddings t_D and t_I to provide predictions. These cross-domain pseudo labels promote generalization and also serve as replay cues. The main objective to map ImageNet data into the downstream label space is for two key reasons; i) it provides an auxiliary supervisory signal that enhances semantic understanding of the downstream classes, ii) it supports privacy-preserving replay by substituting downstream exemplars with semantically aligned ImageNet samples. To achieve this, multiple alignment objectives are designed to ensure consistency across both domains (downstream and ImageNet) and modalities (image and text). The overall loss is defined as

$$\mathcal{L}_{\text{align}} = \mathcal{L}_{DD} + \lambda_1 \mathcal{L}_{II} + \lambda_2 \mathcal{L}_{ID} + \lambda_3 \mathcal{L}_{DI},$$

where each component contributes to a specific alignment:

i) **Downstream-Downstream (D2D) alignment:** This loss uses the DINO-generated pseudo labels and predicted labels from t_D to supervise the CLIP branch for downstream data:

$$\mathcal{L}_{DD} = \mathcal{L}_{\text{CE}}(t_D(f_{I_p}(\text{aug}(x_{D_i}^t))), \phi_D(f_D(x_{D_i}^t))).$$

ii) **ImageNet-ImageNet (I2I) alignment:** This loss uses ImageNet ground-truth labels to train the prompted CLIP features directly:

$$\mathcal{L}_{II} = \mathcal{L}_{\text{CE}}(t_I(f_{I_p}(\text{aug}(x_{I_i}^t))), y_{I_i}^t).$$

iii) **ImageNet-to-Downstream (I2D) alignment:** This key loss term maps ImageNet features into the downstream label space through the downstream adapter:

$$\mathcal{L}_{ID} = \mathcal{L}_{\text{CE}}(t_D(f_{I_p}(\text{aug}(x_{I_i}^t))), \phi_D(f_D(x_{I_i}^t))).$$

In this process, the DINO features of ImageNet images are passed through the downstream adapter ϕ_D to generate pseudo labels in the downstream domain, while the augmented ImageNet images are simultaneously forwarded through the prompted CLIP visual encoder f_{I_p} and projected onto the downstream text embeddings t_D to predict

\mathcal{L}_{DD}	\mathcal{L}_{II}	\mathcal{L}_{DI}	\mathcal{L}_{ID}	\mathcal{L}_{KLD}	\mathcal{L}_{KL_I}	Avg	T1	T2	T3	T4	T5
✓	x	x	x	✓	x	76.572	93.55	86.54	83.95	62.76	56.06
✓	x	x	x	✓	✓	78.522	93.55	86.25	85.63	67.41	59.77
✓	✓	x	x	✓	✓	77.062	93.12	86.98	84.67	65.05	55.49
✓	x	✓	x	✓	✓	76.772	93.12	87.34	83.81	64.70	54.89
✓	x	x	✓	✓	✓	86.312	93.12	85.53	88.76	83.58	80.57
✓	✓	✓	✓	✓	✓	86.348	93.12	85.82	88.71	83.96	80.13

Table 1. Ablation study evaluating the contribution of each loss component in our framework for Oxford Pets dataset. The best results are obtained with all the proposed components.

class labels. Together, these complementary pathways establish a consistent semantic mapping from ImageNet to the downstream label space.

iv) **Downstream-to-ImageNet (D2I) alignment:** This complementary term encourages downstream features to align with ImageNet semantics via the ImageNet adapter:

$$\mathcal{L}_{DI} = \mathcal{L}_{CE}(t_I(f_{I_p}(aug(x_{D_i}^t))), \phi_I(f_D(x_{D_i}^t))).$$

Together, these objectives ensure bidirectional supervision between the downstream and ImageNet domains, allowing the CLIP visual encoder to learn a task-aware and semantically consistent representation. Through prompt tuning of f_{I_p} and joint refinement of \mathbf{t}_D and \mathbf{t}_I , the model successfully bridges the DINO and CLIP spaces, achieving effective annotation-free continual learning while preserving privacy and improving cross-domain generalization.

Stage-5: Replay Strategy. To mitigate catastrophic forgetting, after completing training at task t , we use the prompted CLIP image encoder f_{I_p} to extract visual embeddings for the ImageNet images. Each embedding is then compared against the downstream text prototypes \mathbf{t}_D , and the top- K ImageNet samples per class from \mathcal{A}^t are selected based on similarity. These selected ImageNet images are associated with the corresponding downstream task labels, effectively forming a synthetic replay dataset that is publicly available, privacy-safe, and domain-independent. Because of the previously established cross-domain alignment, these samples are well-aligned semantically and can serve as reliable proxies for replay without storing any task-specific exemplars. In addition to replay, we adopt a temporal knowledge distillation (KD) strategy to stabilize the textual prototypes over time. Following each task, we store the text prototypes \mathbf{t}_D^{t-1} and \mathbf{t}_I^{t-1} from the previous task and regularize the current prototypes \mathbf{t}_D^t and \mathbf{t}_I^t to remain consistent through a Kullback–Leibler (KL) divergence loss:

$$\begin{aligned} \mathcal{L}_{KD} &= KL\left(\text{Softmax}\left(\frac{\mathbf{t}_D^{t-1}}{\tau}\right) \parallel \text{Softmax}\left(\frac{\mathbf{t}_D^t}{\tau}\right)\right) \\ &\quad + KL\left(\text{Softmax}\left(\frac{\mathbf{t}_I^{t-1}}{\tau}\right) \parallel \text{Softmax}\left(\frac{\mathbf{t}_I^t}{\tau}\right)\right), \end{aligned}$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{align}} + \lambda_4 \mathcal{L}_{KD}.$$

where τ is a temperature scaling factor that controls the smoothness of the probability distributions. This loss constrains the evolution of text prototypes across tasks, ensuring that semantic drift is minimized while still allowing adaptation to new classes.

4. Experiments

Baselines. In our experiments, we compare the proposed method against a diverse set of strong baselines spanning four major categories. **CLIP-based baselines:** We compare against Continual CLIP [19], which incrementally adapts the CLIP model over tasks, and CLIP with Pseudo-Label (CLIP-PL) Training, where pseudo labels generated by CLIP are used for supervision. **Unlabeled image recognition baselines:** We evaluate Seq-NoLA and Seq-LaFTer, which adapt NoLA [10] and LaFTer [13] to continual learning by applying their unlabeled image tuning strategy sequentially across tasks. **Test-time tuning baselines:** We evaluate on TPT [17], which adapts CLIP prompts during inference to handle distribution shifts. **Continual Learning baselines:** We benchmark against several state-of-the-art prompt-based and adapter-based methods; L2P [20], DualPrompt [21], CODAPrompt [18], MoEAdapter [23], RAPF [9], and GIFT [22], all of which assume labeled data but have been designed uniquely to handle catastrophic forgetting. Here, instead of labeled data, we use CLIP pseudo labels as proxy labels for the given downstream images. **Datasets:** We conduct evaluations on four challenging datasets representing diverse visual domains which are DTD [4], CIFAR-100 [12], Oxford Pets [15], and Oxford Flowers [14] to assess both generalization and robustness. As there are no prior works, we evaluate on a new benchmark, where we divide the dataset into 5 tasks and distribute the total number of classes equally across 5 tasks. **Metrics:** For evaluation, we compute classification accuracy for each task and report both the task-wise accuracies and the average accuracy across all tasks to measure overall continual performance.

4.1. Implementation Details

We build on CLIP ViT-B/32 with all CLIP and DINO backbones frozen; only (i) T visual prompt tokens, (ii) CLIP LayerNorm scale/bias, and (iii) a lightweight class adapter (Linear 512→C, no bias) are trainable. Class prototypes are computed by averaging L2-normalized text embeddings from dataset-specific prompts and “a photo of a {}.” used both to initialize the classifier and to retrieve up to three nearest ImageNet synsets per class. Replay stores 10 images per semantically matched synset and runs via a small ImageNet loader alongside downstream data. We train with AdamW (lr 0.004, betas 0.9/0.999, wd 0.01) and step-decay 0.2; batch sizes are 256 (downstream), 32 (top-k warm-up), and 64 (replay, 4 workers). Losses are

Method	DTD					CIFAR-100					Oxford Pets					Oxford Flowers					Overall Avg				
	Avg	T1	T2	T3	T4	T5	Avg	T1	T2	T3	T4	T5	Avg	T1	T2	T3	T4	T5	Avg	T1	T2	T3	T4	T5	
CLIP-based																									
Continual CLIP	47.04	65.0	50.83	42.43	39.47	37.47	48.38	64.68	53.36	47.08	39.37	37.41	78.47	90.11	76.34	76.26	73.83	75.82	64.00	80.91	67.95	58.04	57.41	55.70	59.47
CLIP-PL	46.43	75.28	56.81	24.62	40.28	35.17	64.46	93.32	74.92	56.21	48.48	49.38	73.94	94.41	88.86	82.41	53.35	50.67	50.00	88.32	50.50	42.97	30.94	37.27	58.21
Unlabeled Image Recog.																									
Seq LaFTer (CVPR'24)	35.02	78.06	37.78	24.33	19.23	15.72	36.35	90.21	40.44	26.94	12.96	11.18	40.49	91.98	45.59	28.06	20.41	16.43	33.48	88.60	40.06	16.71	14.82	7.23	36.84
Seq NoLA (ArXiv'24)	54.54	86.38	77.08	50.57	32.89	25.80	45.12	93.31	58.56	26.86	26.57	20.32	62.80	94.27	82.52	53.78	45.81	37.66	67.12	94.87	79.68	58.35	55.68	47.05	57.90
Test-Time Tuning																									
TPT (ICLR'23)	25.83	40.28	23.75	21.07	21.42	22.64	58.44	72.21	66.15	55.94	50.45	47.49	69.46	74.64	66.43	72.66	65.43	68.17	30.04	47.86	28.61	26.42	25.44	21.88	45.94
Continual Learning																									
L2P (CVPR'22)	23.85	57.50	31.39	15.71	9.28	5.38	23.71	86.95	10.64	8.32	8.08	4.58	31.88	72.78	38.78	22.68	15.10	10.06	23.16	60.40	29.18	13.75	5.88	6.37	25.65
DualPrompt (ECCV'22)	31.27	72.78	35.42	20.02	15.50	12.65	23.48	87.68	14.51	5.74	5.71	3.79	36.49	80.80	40.67	27.10	19.40	14.50	31.10	73.50	37.48	19.89	14.77	9.87	30.59
CODAPrompt (CVPR'23)	37.63	80.83	40.69	28.16	21.27	17.20	38.47	83.11	40.49	29.79	21.63	17.35	37.19	81.66	41.24	27.39	19.78	15.89	37.13	88.32	44.35	24.01	16.55	12.42	37.10
MoEAdapter (CVPR'24)	30.61	88.89	31.81	16.38	10.31	5.67	17.52	38.89	20.90	11.89	10.89	5.06	6.49	14.33	7.16	4.76	3.47	2.73	37.99	89.17	24.46	40.09	22.53	13.72	23.15
RAPF (ECCV'24)	53.66	79.17	57.64	47.32	43.35	40.84	58.04	82.89	63.79	54.06	46.58	42.92	80.74	92.69	79.09	78.42	75.81	77.71	69.03	91.97	73.68	61.69	60.32	57.53	65.37
GIFT (CVPR'25)	51.11	88.06	63.19	45.59	30.12	28.61	57.32	93.74	65.00	45.94	47.62	34.34	52.87	91.83	63.17	44.31	34.85	30.23	57.23	95.73	65.09	57.26	38.27	29.84	54.13
Ours																									
CrossWorld-CL	62.65 (+8.11)	84.17	74.31	57.85	51.10	45.80	67.44 (+2.98)	93.11	79.85	66.60	51.45	46.19	86.35 (+5.60)	93.12	85.82	88.71	83.96	80.16	71.21 (+2.18)	94.02	80.54	63.95	60.86	56.68	71.91 (+6.54)

Table 2. **Main Result:** We compare our method with different methods namely CLIP baselines, unlabeled image recognition methods, test-time tuning methods, and continual learning approaches across four datasets: DTD, CIFAR-100, Oxford Pets, and Oxford Flowers. For each method, we report per-task accuracy (Task-1 to Task-5), dataset-level averages, and the overall average across all benchmarks. Our approach achieves the highest overall accuracy, with consistent gains on all datasets, particularly on later tasks highlighting the benefit of using auxiliary world knowledge and cross-domain alignment to improve robustness and reduce forgetting in the annotation-free continual learning setting. All results are reported in %.

cross-entropy in downstream and ImageNet spaces and λ_1 , λ_2 , $\lambda_3=1$ and $\lambda_4 = 30$. Trainable parameters are $768T + 512C + 39,936$, where T is the number of prompt tokens (default 16), C is the number of downstream classes, and 39,936 is the number of LayerNorm parameters in CLIP ViT-B/32, i.e., for CIFAR-100 103k; Oxford Pets 71k; DTD 76k; Oxford Flowers 104k which is hardly 0.047–0.069% of the total trainable parameters (151M) of ViT-B/32. Replay remains tiny: by Task 5 the union of ImageNet proxy classes is 122 (Pets), 87 (DTD), 219 (Flowers), 344 (CIFAR-100), yielding only 870–3,440 images (Approx 0.00068–0.00269 fraction of the 1.28M-image ImageNet train set). Experiments use PyTorch 2.4.1/CUDA 12.2 on a Tesla V100-SXM2-32GB. We will make our code publicly available upon acceptance.

4.2. Main Results

The results presented in Table 2, show a consistent and significant advantage of our method across all four datasets compared to CLIP-based, Unlabeled image recognition, test-time tuning, and continual learning baselines. CLIP-only methods struggle as the number of classes grows, particularly in the later tasks (Task-3, Task-4, and Task-5), where class boundaries become finer and the visual variations between categories increase. Unlabeled image recognition baselines such as Seq NoLA and Seq LaFTer improve the early-task accuracy but degrade sharply in later tasks due to error accumulation in pseudo labeling and the lack of mechanisms to handle distribution drift. Classical continual learning approaches (L2P [20], DualPrompt [21], CODAPrompt [18], MoEAdapter [23], RAPF [9], GIFT [22]), although strong under supervised settings, collapse in this

annotation-free scenario because they rely heavily on labeled exemplars and fail to maintain discriminative features without them. In contrast, our method achieves the highest average performance on every dataset, with especially large margins on Oxford Pets and Oxford Flowers, where fine-grained distinctions make the task more challenging. A key observation is that our gains are not limited to early tasks but become more pronounced in later tasks. This demonstrates that the auxiliary world knowledge from ImageNet stabilizes learning as new classes arrive, providing strong semantic anchors that prevent confusion among visually similar categories.

4.3. Further Analysis

Ablation Study. To better understand the contribution of each loss component, we systematically enable and disable them and study the effect on all task accuracies as shown in Table 1. Firstly, when either of the cross-domain alignment losses (\mathcal{L}_{DI} or \mathcal{L}_{ID}) is removed, the model loses its ability to reliably connect downstream samples with the auxiliary ImageNet supervision. This doesn't completely effect the early tasks, where the label space is still small, but it becomes much more noticeable on the later tasks starting from Task-3. The drop in accuracy in these settings suggests that the model increasingly relies on the auxiliary supervision to keep class boundaries stable as the task sequence grows. Without these losses, the downstream and ImageNet spaces drift apart, which makes the model less confident and more error-prone as new classes arrive. We also observe that the KL consistency losses have an important stabilizing effect. When they are removed, the performance on Task-1 and Task-2 remains mostly unchanged as there is not much

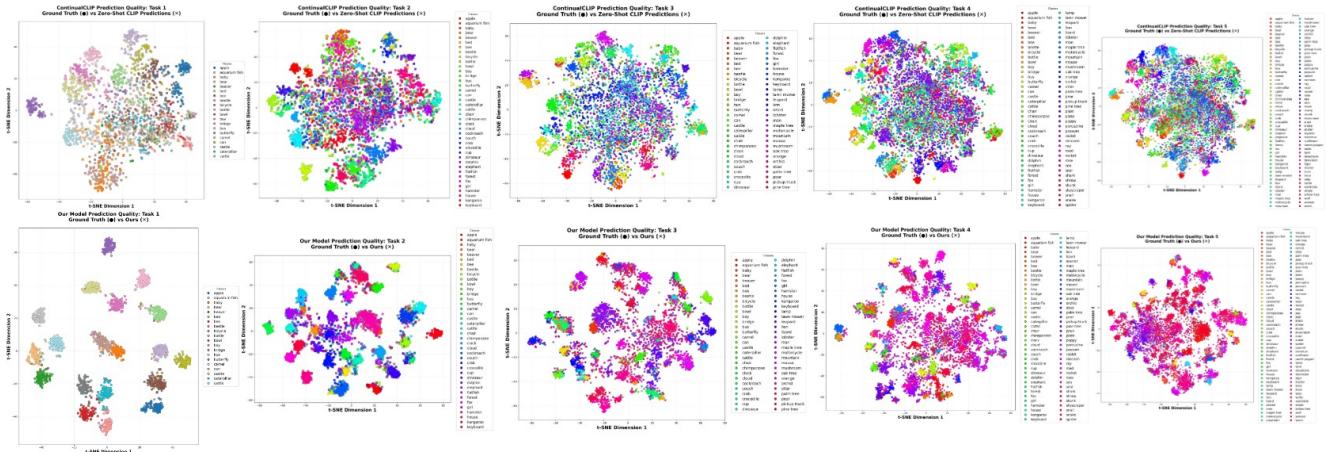


Figure 4. t-SNE visualizations comparing prediction quality of Continual CLIP (top row) and our method (bottom row) across all five tasks for CIFAR100 dataset. Continual CLIP produces highly entangled clusters with significant overlap between classes, indicating poor separation and increasing confusion as tasks progress. In contrast, our model yields well formed, compact, and clearly separated clusters for every task, showing that cross domain alignment and world knowledge driven supervision lead helps.

change, but the later tasks degrade. This shows that the KL terms act as a temporal regularizer, keeping the predictions from drifting too far between tasks and helping the model preserve the structure it learned earlier. Other configurations that use only a subset of these losses do provide some benefit, but they tend to emphasize one aspect of the problem at the expense of another. For example, relying only on downstream supervision biases the model toward task-specific pseudo labels, while relying only on ImageNet alignment does not fully capture the semantics of the downstream distribution. Neither approach alone is sufficient to maintain stable performance across tasks. The strongest results are consistently obtained when all components which includes alignment losses and KL consistency losses are used together proving the merit of our proposed method.

Effect of k and Auxiliary Images. Figure 3 illustrates how accuracy changes as we vary k , the number of ImageNet classes retrieved per downstream class, and the number of auxiliary images sampled from each of those classes. The performance on the early tasks, Task-1 to Task-3 remains relatively stable across all settings, suggesting that the model does not require a large amount of auxiliary support when the downstream label space is still small. The impact of k and image count becomes much clearer in the later tasks; Task-4 and Task-5. As more classes are introduced, the downstream data becomes more ambiguous, and the model benefits from a richer auxiliary set: increasing k provides a broader semantic neighborhood, while using more images per class supplies stronger visual evidence. Together, these factors lead to consistent improvements on the more challenging later tasks, helping the model disambiguate visually similar classes in this setting.

Qualitative Analysis using tSNE plots. Figure 4 shows t-SNE visualizations of Continual CLIP (top row) and our method (bottom row) across all tasks, with ground-truth and predicted labels overlaid. Continual CLIP exhibits progressively increasing cluster overlap, with many classes collapsing into each other as tasks advance, clear evidence of representation drift and forgetting. In contrast, our method produces tighter, well-separated clusters with much closer alignment between ground truth and predictions. The structure of earlier classes is preserved even in later tasks, demonstrating stable representations and reduced confusion. These visualizations confirm that the combination of auxiliary ImageNet supervision and cross-domain alignment leads to a more robust and discriminative feature space compared to Continual CLIP.

5. Conclusion

We introduced an annotation free class incremental learning framework that uses world knowledge as a stable external supervisor to guide learning in the absence of labels. By aligning downstream representations with semantically related ImageNet classes and designing a cross domain training strategy that connects CLIP, DINO, and auxiliary supervision, our approach maintains coherent features across tasks while avoiding the need to store any downstream exemplars. Extensive experiments across four challenging datasets show that our method consistently outperforms CLIP based, test time adaptation based, and continual learning baselines, with especially large gains on later tasks where class confusion is highest. The qualitative analyses further demonstrate that our model retains clearer decision boundaries and suffers far less drift over time. Overall, our results highlight the value of using structured external knowledge to support continual learning when annotations

are unavailable, and point towards a promising direction.

References

- [1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- [2] Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021.
- [3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020.
- [4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2013.
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.
- [6] Jean-Bastien Grill, Florian Strub, Florent Altch'e, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *ArXiv*, abs/2006.07733, 2020.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2019.
- [9] Linlan Huang, Xusheng Cao, Haori Lu, and Xialei Liu. Class-incremental learning with clip: Adaptive representation adjustment and parameter fusion. In *European Conference on Computer Vision*, 2024.
- [10] Mohamed Fazli Mohamed Imam, Rufael Marew, Jameel Hassan, Mustansar Fiaz, Alham Fikri Aji, and Hisham Cholakkal. Clip meets dino for tuning zero-shot classifier using unlabeled image collections. *ArXiv*, abs/2411.19346, 2024.
- [11] Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El-Saddik, and Eric P. Xing. Efficient test-time adaptation of vision-language models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14162–14171, 2024.
- [12] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical Report TR-2009, University of Toronto, 2009.
- [13] Muhammad Jehanzeb Mirza, Leonid Karlinsky, Wei Lin, Mateusz Koziński, Horst Possegger, Rogério Schmidt Feris, and Horst Bischof. Lafter: Label-free tuning of zero-shot classifier using language and unlabeled image collections. *ArXiv*, abs/2305.18287, 2023.
- [14] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008.
- [15] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012.
- [16] 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 *International Conference on Machine Learning*, 2021.
- [17] 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. *ArXiv*, abs/2209.07511, 2022.
- [18] James Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogério Schmidt Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11909–11919, 2022.
- [19] Vishal Thengane, Salman A. Khan, Munawar Hayat, and Fahad Shahbaz Khan. Clip model is an efficient continual learner. *ArXiv*, abs/2210.03114, 2022.
- [20] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. Learning to prompt for continual learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 139–149, 2021.
- [21] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. Dualprompt: Complementary prompting for rehearsal-free continual learning. *ArXiv*, abs/2204.04799, 2022.
- [22] Bin Wu, Wuxuan Shi, Jinqiao Wang, and Mang Ye. Synthetic data is an elegant gift for continual vision-language models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2813–2823, 2025.
- [23] Jiazu Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning

of vision-language models via mixture-of-experts adapters.
2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 23219–23230, 2024.

Annotation-Free Class-Incremental Learning

Supplementary Material

This supplementary material contains additional details that we could not include in the main paper due to space constraints, including the following information.

- Discussion on Methodology and Design Choices in Sec 6.
- Ablation Study on Replay Samples per class is shown in Sec 7.
- Forgetting Measure Study in Sec 8.
- Replacing the DINO Backbone is studied in Sec 9.
- Qualitative Samples from World-Knowledge Distillation is presented in Sec 10.
- Algorithm of the overall proposed method is presented in Sec 11.
- Prompts used to query LLM are presented in Sec 12.

6. Discussion on Methodology and Design Choices.

In this section we reflect on key methodological choices and address potential concerns that may arise when using ImageNet samples as auxiliary exemplars in an Annotation-free Continual Learning setting. One could wonder that whether adding such external data still qualifies as continual learning or whether it injects undesired supervision into the process. We emphasize that continual learning restricts access only to past task data, not to publicly available external sources. ImageNet samples do not belong to any task in the incremental stream and remain fixed throughout training. Their inclusion therefore does not break the sequential nature of the problem. Using them is conceptually identical to using a pretrained backbone, which every continual learning baseline already depends on. We also note that recent state of the art work such as GIFT [22] employs synthetic ImageNet imagery generated through diffusion models and leverages auxiliary supervision even when labeled data is available. This further reinforces the motivation for incorporating auxiliary supervision in our setting, since leading continual learning methods already demonstrate the effectiveness and acceptance of such external structure. Finally, one might wonder whether the use of ImageNet makes the problem artificially easier or produces an unfair comparison. In practice, all VLM based methods already rely on massive pretrained priors, often orders of magnitude larger and richer than ImageNet. We make use of ImageNet to learn from its inherent semantic knowledge. Our use of a small, fixed set of publicly available proxy exemplars is a considerably weaker prior and provides a transparent and reproducible source of world level structure. This reflects a principled design choice: instead of relying on hidden, web scale training data or proprietary multimodal LLM supervi-

<i>k</i>	Avg	T-1	T-2	T-3	T-4	T-5
1	85.67	92.84	86.11	85.30	83.76	80.38
2	85.69	92.84	85.31	89.09	81.67	79.58
5	84.15	92.84	85.38	85.30	80.08	77.19
10	86.29	92.84	85.82	88.71	83.96	80.13

Table 3. Ablation on the number of auxiliary replay samples per class (*k*) on the Oxford Pets dataset. The results show that performance remains stable across a wide range of replay sizes, indicating that our method is not highly sensitive to the exact number of ImageNet samples used.

sion, we adopt a controlled and publicly accessible auxiliary source to stabilize pseudo labels without breaking the core constraints of annotation free continual learning.

7. Ablation Study on Replay Samples per Class

The ablation on the Oxford Pets dataset in Table 3 shows that our method is not highly sensitive to the number of auxiliary ImageNet samples used during replay. Even when varying the replay size from $k = 1$ to $k = 10$ samples per class, the overall performance remains stable, with average accuracy staying within a narrow band of 84–86%. This indicates that the distilled ImageNet exemplars provide strong supervisory value even in very small quantities. The model benefits consistently from the semantic structure captured in these auxiliary images, and does not rely on large replay buffers. This stands in contrast to traditional continual learning methods that require storing around 20 real exemplars per class, which introduces storage overhead and privacy concerns. In our case, only a handful of public, non private ImageNet samples are sufficient, demonstrating both efficiency and robustness in the annotation-free setting.

8. Forgetting Measure Study

To quantify the degradation of previously learned knowledge as new tasks arrive, we adopt the standard *forgetting measure* used in continual learning. For a given task t , forgetting is defined as the drop between the best accuracy ever achieved on that task and its final accuracy after completing all tasks. Formally, the forgetting for task t is computed as

$$F(t) = \max_{k \leq T} a_t^{(k)} - a_t^{(T)},$$

where $a_t^{(k)}$ denotes the accuracy on task t after training on task k , and T is the index of the final task. Larger values

indicate more severe forgetting, while negative values imply improvement over time.

Across the four datasets, the forgetting patterns exhibit distinct characteristics as shown Figure 5. DTD shows mild fluctuations, with forgetting values roughly between 1.5% and 2.2% across tasks even with a very distinct domain gap between the distilled knowledge and the downstream task, indicating stable retention with modest performance drops. CIFAR-100 displays stronger forgetting, particularly in later tasks where values exceed 10, reflecting the dataset’s higher diversity and difficulty. Despite this, our method still outperforms all baselines by a margin of 3%. Oxford Pets exhibits an interesting pattern: the early tasks show negative forgetting, indicating performance gains, while the later tasks experience only mild positive forgetting, with values ranging from -2.97% to $+0.73\%$. Notably, the improvement contributed by the auxiliary data (approximately 3%) substantially outweighs the minimal forgetting effect (around 0.7%), rendering the latter practically negligible. Oxford Flowers exhibits a mixed trajectory, where forgetting increases around task Task-2 but subsequently decreases and becomes slightly negative at Task-4, indicating partial recovery after mid sequence degradation. These trends highlight how forgetting behaves differently depending on dataset characteristics, granularity, and inter task relationships. Overall, these results show that the influence of forgetting is minimal compared to the gains achieved, confirming the robustness of our method across all datasets.

9. Replacing the DINO Backbone

The backbone ablation in Table 4 demonstrates that our method remains effective across a wide range of architectures, including lightweight models with significantly fewer parameters. Notably, when replacing the DINO 16-bit backbone (86M params) with the much smaller ViT-Tiny model (only 5.7M params), the performance remains remarkably stable. Despite being more than an order of magnitude smaller, ViT-Tiny achieves an overall average of 69.97%, outperforming the best prior SOTA by a margin of +4.60%. Furthermore, the dataset-wise improvements remain consistent, showing strong gains on Oxford Flowers and Oxford Pets, and competitive results on DTD and CIFAR-100. This indicates that the benefits of our auxiliary ImageNet-based distilled supervision are not tied to a specific architecture capacity; instead, the method generalizes reliably even with compact models. The strong performance of such a lightweight backbone highlights the efficiency and robustness of our approach, suggesting that the gains stem primarily from the quality of auxiliary supervision rather than the scale of the model. We also show a performance gain of +7.47% on DINO 8-bit backbone (21M params) which is slightly bigger than ViT-Tiny, making the method more robust across different backbones.

10. Qualitative Samples from World-Knowledge Distillation

The examples in Figure 6 illustrate how world knowledge from ImageNet can be distilled to support downstream classes in an Annotation-Free Continual Learning paradigm. For each downstream class (shown in blue), we retrieve a small set of semantically related ImageNet classes, displayed adjacent to it. These retrieved concepts consistently reveal meaningful visual or semantic correspondences: for instance, CIFAR-100’s Kangaroo retrieves ImageNet classes containing similar animal poses and textures, while Oxford Flowers’ Globe Flower retrieves spherical or patterned objects such as Soccer Ball and Balloon, capturing strong shape-based similarity. Likewise, in DTD, texture classes such as Knitted or Polka-dotted retrieve ImageNet samples exhibiting similar structural repetition, color distribution, and geometric regularity. These relationships highlight that ImageNet contains a rich semantic prior that can be repurposed, even without explicit labels as auxiliary supervision for a wide range of downstream categories.

The retrieved samples thus act as proxy exemplars that encode the high-level concepts associated with each downstream class, providing the model with stable semantic anchors across tasks. Because they capture discriminative properties such as texture, shape, object geometry, or fine-grained appearance cues, they help reinforce and stabilize representations during continual learning, particularly when the downstream data arrive unlabeled. Importantly, these ImageNet exemplars remove the need to store any real user or task-specific images, addressing long-standing concerns around privacy and data retention in continual learning. By leveraging publicly available world-knowledge images that are semantically mapped to downstream classes, our method delivers a principled and privacy-preserving replay mechanism while still achieving substantial performance improvements across all tasks.

11. Algorithm

Algorithm-1 provides an overview of our Annotation-Free Continual Learning pipeline, where each stage contributes a complementary form of structure to stabilize learning across tasks. At a high level, the algorithm first retrieves relevant world knowledge from ImageNet, expands the downstream task knowledge semantically using an LLM, aligns visual and textual spaces using dual supervision by mapping DINO to CLIP space, further enforces cross-domain consistency through prompt-guided alignment, and finally selects a compact replay set to guide future tasks. Each stage has a clear intuition: Stage-1 identifies ImageNet classes that are semantically closest to the downstream labels, giving the model a “world-knowledge prior” that anchors its representations (**Algorithm-2**). Stage-2 enriches downstream text

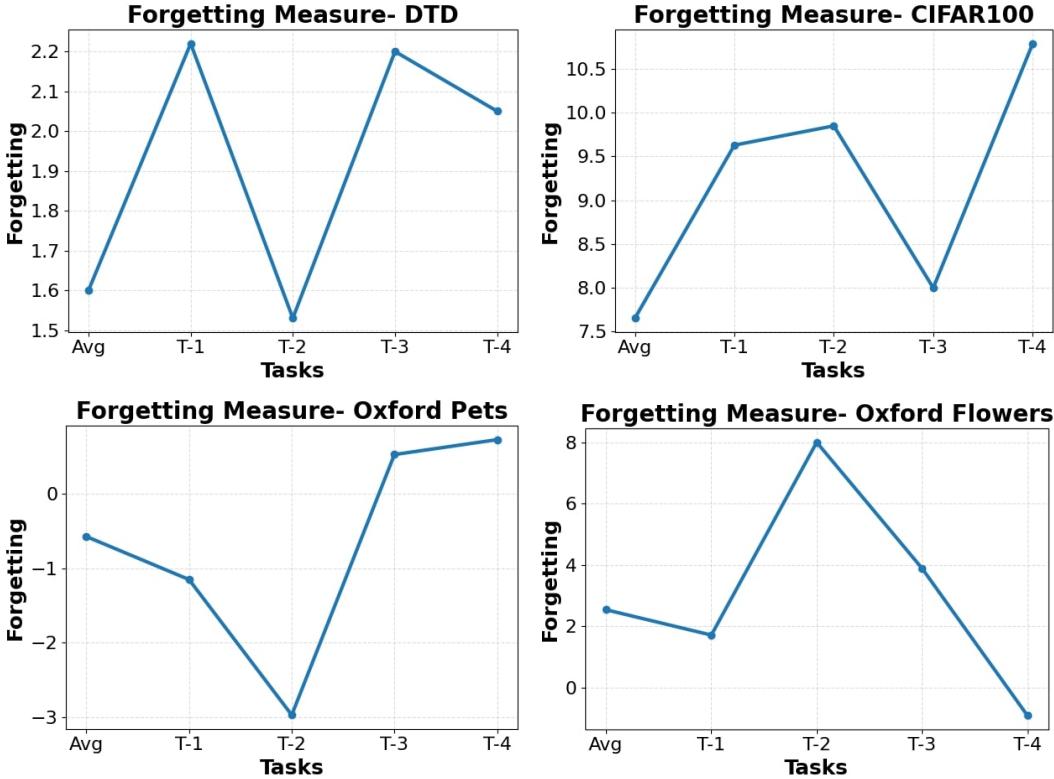


Figure 5. We report forgetting values for each task on DTD, CIFAR-100, Oxford Pets, and Oxford Flowers. The trends highlight dataset specific forgetting dynamics, with CIFAR-100 showing higher forgetting, Oxford Pets exhibiting negative forgetting in early tasks, and Oxford Flowers and DTD displaying moderate fluctuations.

Backbone	Dataset	Overall Avg					
			Avg	T-1	T-2	T-3	T-4
Best-performing SOTA (RAPF (ECCV'24))	DTD	65.37	53.66	79.17	57.64	47.32	43.35
	CIFAR100		58.04	82.89	63.79	54.06	46.58
	Oxford Pets		80.74	92.69	79.09	78.42	75.81
	Oxford Flowers		69.03	91.97	73.68	61.69	60.32
DINO 16-bit (86M) (CrossWorld-CL from Table [2] of main paper)	DTD	71.912 (+6.54)	62.658	84.17	74.31	57.85	51.10
	CIFAR100		67.440	93.11	79.85	66.60	51.45
	Oxford Pets		86.340	93.12	85.82	88.71	83.96
	Oxford Flowers		71.210	94.02	80.54	63.95	60.86
ViT-Tiny (5.7M)	DTD	69.972 (+4.60)	62.656	80.28	72.64	58.81	53.80
	CIFAR100		61.922	89.53	75.46	62.70	48.47
	Oxford Pets		83.550	87.54	82.56	87.17	81.05
	Oxford Flowers		71.760	94.59	81.83	64.65	60.38
DINO 8-bit (21M)	DTD	72.8485(+7.47)	65.106	84.44	76.67	61.02	53.87
	CIFAR100		69.872	92.53	80.13	68.85	58.85
	Oxford Pets		83.530	89.54	84.80	88.60	78.06
	Oxford Flowers		72.886	94.87	83.26	67.52	61.29

Table 4. Replacing DINO with lighter backbones such as ViT-Tiny or 8-bit DINO still provides strong and consistent performance, indicating backbone independence. The table reports task-wise results (Avg and Task-1 to Task-5) and the Overall Avg for each backbone. The gains in blue are reported against the best performing SOTA method.

Oxford Pets

CIFAR-100



DTD

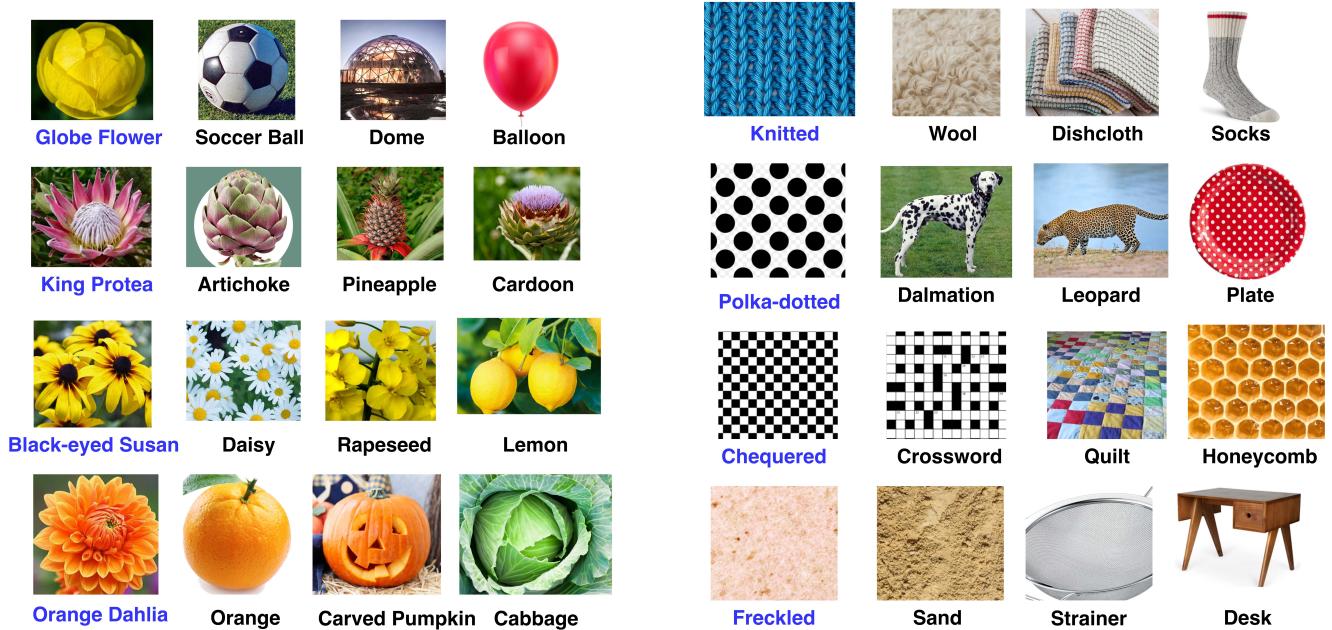


Figure 6. Qualitative examples of distilled world knowledge. For each downstream class (in blue), we show semantically related ImageNet images retrieved through our distillation step. These examples highlight how auxiliary world knowledge provides meaningful visual cues that align with downstream classes, enabling privacy-safe and effective guidance for annotation-free continual learning.

embeddings by using an LLM to provide richer, more descriptive semantics (**Algorithm-3**), helping the model reason at a concept level rather than from short label tokens. Stage-3 then aligns both downstream and auxiliary images from DINO visual space to CLIP text spaces (**Algorithm-4**), ensuring that visual and linguistic semantics reinforce each other by training two adapters. The trained DINO adapters act as a pseudo-labeler to train the prompted CLIP model in next stage. The reason DINO is chosen because of its ability to capture object-centric features that later help the model in cross-domain alignment. Stage-4 enforces cross-domain alignment between CLIP and DINO, allowing both encoders to agree on a shared semantic space and preventing drift as new tasks arrive (**Algorithm-5**). This way the prompted CLIP model learns a generic representation using DINO and the auxiliary data with the help of all the alignment losses. Finally, Stage-5 distills the auxiliary pool into a small set of high-confidence exemplars using prompted CLIP model scores (**Algorithm-6**), forming a replay memory that is privacy-safe, compact, and semantically meaningful. Together, these stages create a stable learning framework where external world knowledge reduces noise in pseudo labels, alignment losses tie representations together across domains, and replay ensures long-term retention across tasks. We also clarify that the auxiliary ImageNet data is introduced only from Task-2 onward. Since Task-1 contains very few classes, we avoid adding auxiliary samples at this stage to prevent the model from being confused early in training. We also clarify that we don't use any task-id during inference.

12. Prompts used to query LLM

In Stage 2 we query an large language model (GPT 3.5) with class names to obtain richer semantic descriptions of each downstream category. For every class label, we instantiate a small set of dataset specific prompt templates and send only the text label (no images) to the LLM. The returned descriptions include attributes, shapes, materials, typical contexts, and synonyms. We then concatenate these descriptions with the original label and encode the resulting text using the CLIP text encoder to form the task aware embeddings t_D . The same descriptions are also used to refine the mapping between downstream labels and candidate ImageNet classes. In this way, Stage 2 injects high level semantic knowledge into the text space while remaining annotation free: the LLM is never asked to label images, but only to explain what each class usually looks like in a realistic photograph that matches the dataset domain.

DTD (textures) prompts.

- Describe a close up texture of {category}.
- How does the {category} texture

usually look in a photograph?

- What visual pattern and material properties define the {category} texture?
- How can you recognize {category} from a small image patch?
- Describe a photo that clearly shows the {category} texture.

CIFAR-100 prompts.

- Describe a natural photo of {category}.
- How does {category} usually appear in a real world image?
- What visual details help you identify {category} in a picture?
- Describe a typical scene that contains {category}.
- How can you recognize {category} when looking at a single image?

Oxford Pets prompts.

- Describe a portrait photo of a {category} pet.
- How does a {category} cat or dog usually look in a photo?
- What facial features and fur patterns are typical for a {category}?
- How can you identify a {category} in a pet photograph?
- Describe a clear photo that shows a {category} from the front.

Oxford Flowers prompts.

- Describe a close up photo of a {category} flower.
- What are the typical color, shape, and structure of a {category} blossom?
- How does a garden photo of {category} flowers usually look?
- How can you recognize a {category} in a photograph of plants?
- Describe a detailed image that focuses on a single {category} bloom.

Algorithm 1 CrossWorld-CL for Annotation Free Continual Learning Paradigm

Require: Stream of tasks $\{D^t\}_{t=1}^T$ where $D^t = \{X_D^t, Y_D^t\}$

Require: ImageNet auxiliary pool \mathcal{X}_I ; CLIP encoders f_I (image), f_t (text); DINO backbone f_D

Require: LLM; replay budget k (top k samples per class)

Ensure: Trained adapters ϕ_D, ϕ_I , llm-prompts, and replay memory \mathcal{A}_R

- 1: Initialize adapters ϕ_D, ϕ_I and prompts randomly
- 2: Initialize global replay memory $\mathcal{A}_R \leftarrow \emptyset$
- 3: **for** $t = 1$ to T **do**
- Stage 1: Task Aware World Knowledge Distillation**
- 4: $\mathcal{A}^t \leftarrow$ World Knowledge Distillation $(Y_D^t, \mathcal{X}_I, \mathcal{Y}_I, f_t)$
- Stage 2: Semantic Expansion through LLM**
- 5: $t_D, t_I \leftarrow$ Semantic Expansion $(\mathcal{Y}_D^t, LLM, \mathcal{A}^t, f_t, \text{llm_prompts})$
- Stage 3: Dual Supervised Visual Semantic Alignment**
- 6: $\phi_D, \phi_I \leftarrow$ Dual Supervised Alignment $(\mathcal{X}_D^t, \mathcal{A}^t, t_D, f_I, f_D, \phi_D, \phi_I)$
- Stage 4: Prompt Guided Cross Domain Alignment**
- 7: $f_{I_p}, t_D, t_I \leftarrow$ Cross Domain Alignment $(\mathcal{X}_D^t, \mathcal{A}^t, t_D, t_I, f_{I_p}, f_D, \phi_D, \phi_I)$
- Stage 5: Replay Strategy**
- 8: $\mathcal{A}_R \leftarrow$ Build Replay $(\mathcal{A}^t, f_{I_p}, t_D)$
- 9: $A^R \leftarrow A^R \cup A_R^t$
- 10: **end for**
- 11: **return** t_D, f_{I_p}
- Inference on test data**
- 12: **for** each test image $x \in \mathcal{T}_{test}^t$ **do**
- 13: $v \leftarrow \text{norm}(f_{I_p}(x))$
- 14:
- 15: **for all** $c \in \mathcal{C}$ **do** ▷ \mathcal{C} : all downstream classes seen so far
- 16: $s[c] \leftarrow \cos(v, t_D[c])$
- 17: **end for**
- 18: $\hat{y}(x) \leftarrow \arg \max_{c \in \mathcal{C}} s[c]$
- 19: **end for**
- 20: **return** predictions $\hat{y}(x)$ for all $x \in \mathcal{T}_{test}^t$

Algorithm 2 Stage 1: World Knowledge Distillation

1: **function** WORLD KNOWLEDGE DISTILLATION($Y_D^t, \mathcal{X}_I, \mathcal{Y}_I, f_t$)

Require: Downstream class names Y_D^t ; ImageNet images \mathcal{X}_I and labels \mathcal{Y}_I ; text encoder f_t

Ensure: Task specific auxiliary pool \mathcal{A}^t

- 2: $\mathcal{A}^t \leftarrow \emptyset$ ▷ Encode downstream and ImageNet label texts
- 3: **for each** $c \in Y_D^t$ **do**
- 4: $t_D^{\text{raw}}[c] \leftarrow \text{norm}(f_t(c))$
- 5: **end for**
- 6: **for each** $u \in \mathcal{Y}_I$ **do**
- 7: $t_I^{\text{all}}[u] \leftarrow \text{norm}(f_t(u))$
- 8: **end for** ▷ Find ImageNet labels closest to each downstream label
- 9: **for each** $c \in Y_D^t$ **do**
- 10: $\mathcal{S}_c \leftarrow \text{Topk}_{u \in \mathcal{Y}_I}(\cos(t_D^{\text{raw}}[c], t_I^{\text{all}}[u]))$
- 11: **for each** $u \in \mathcal{S}_c$ **do**
- 12: $\mathcal{A}^t \leftarrow \mathcal{A}^t \cup \{x \in \mathcal{X}_I \mid \text{label}(x) = u\}$
- 13: **end for**
- 14: **end for**
- 15: **return** \mathcal{A}^t
- 16: **end function**

Algorithm 3 Stage 2: Semantic Expansion through LLM

```

1: function SEMANTIC EXPAN-
   SION( $\mathcal{Y}_D^t$ , LLM,  $\mathcal{A}^t$ ,  $f_t$ , llm_prompts)
Require:  $\mathcal{Y}_D^t$ ; LLM;  $\mathcal{A}^t$ ; text encoder  $f_t$ ; llm_prompts
Ensure: Downstream and ImageNet text embeddings  $t_D, t_I$ 
2:    $t_D \leftarrow \emptyset, t_I \leftarrow \emptyset$ 
3:   for each class  $c \in \mathcal{Y}_D^t$  do
4:     promptc  $\leftarrow$  build prompt from  $c$  and llm_prompts
5:     {descc(1), ..., descc(m)}  $\leftarrow$  query LLM with promptc
         ▷ Compute multiple text embeddings and average them
6:     Initialize list  $\mathcal{E}_c \leftarrow \emptyset$ 
7:     for  $j = 1$  to  $m$  do
8:        $\mathcal{E}_c \leftarrow \mathcal{E}_c \cup \left\{ \text{norm}\left(f_t(\text{desc}_c^{(j)})\right) \right\}$ 
9:     end for
10:     $t_D[c] \leftarrow \frac{1}{m} \sum_{e \in \mathcal{E}_c} e$ 
         ▷ Collect ImageNet text embeddings for selected classes in  $\mathcal{A}^t$ 
11:    for each  $u \in \mathcal{Y}_I$  do
12:      if  $u$  in  $\mathcal{A}^t$  then
13:         $t_I[u] \leftarrow \text{norm}(f_t(u))$ 
14:      end if
15:    end for
16:  end for
17:  return  $t_D, t_I$ 
18: end function

```

Algorithm 4 Stage 3: Dual Supervised Visual Semantic Alignment

```

1: function DUAL SUPERVISED ALIGN-
   MENT( $\mathcal{X}_D^t, A^t, t_D, f_I, f_D, \phi_D, \phi_I$ )
Require: Current task images  $X_D^t$ ; auxiliary dataset  $A^t$ 
Require: Downstream and ImageNet Text embeddings  $t_D, t_I^t$ 
Require: Encoders  $f_D$  (DINO),  $f_I$  (CLIP image); adapters  $\phi_D, \phi_I$ 
Ensure: Updated adapters  $\phi_D, \phi_I$ 

Step 1: Obtain pseudo labels for downstream and replay images
2:   for each  $x \in X_D^t$  do
3:      $z \leftarrow \text{norm}(\phi_D(f_D(x)))$ 
4:      $\hat{y}_{D_i}^t \leftarrow \arg \max_c \cos(z, t_D[c])$ 
5:   end for

Step 2: Train adapters with dual supervision
6:   for each training step do
7:     Sample mini batches  $B_D \subset X_D^t$  and  $B_I \subset A^t$ 
     ▷ Aligning Downstream images with downstream text
8:      $z_D \leftarrow \text{norm}(\phi_D(f_D(B_D)))$ 
9:      $L_{map_1} = \mathcal{L}_{CE}(z_D, \hat{y}_{D_i}^t)$ 
     ▷ Aligning Auxiliary ImageNet images with ImageNet text
10:     $z_I \leftarrow \text{norm}(\phi_I(f_D(B_I)))$ 
11:     $L_{map_2} = \mathcal{L}_{CE}(z_I, y_{I_i}^t)$ 
12:     $L_{map} \leftarrow L_{map_1} + L_{map_2}$ 
13:    Update  $\phi_D, \phi_I$  using gradient of  $L_{map}$ 
14:  end for
15:  return  $\phi_D, \phi_I$ 
16: end function

```

Algorithm 5 Stage 4: Prompt Guided Cross Domain Alignment

```

1: function CROSS DOMAIN ALIGN-
MENT( $\mathcal{X}_D^t, \mathcal{A}^t, t_D, t_I, f_{I_p}, f_D, \phi_D, \phi_I$ )
Require: Downstream images  $\mathcal{X}_D^t$ ; auxiliary pool  $\mathcal{A}^t$ 
Require: Text embeddings  $t_D, t_I$ ; prompted CLIP  $f_{I_p}$ ;
    DINO  $f_D$ ; adapters  $\phi_D, \phi_I$ 
Ensure: Updated  $f_{I_p}, t_D, t_I$ 
2:   for each training step do
3:     Sample mini batch  $B_D, B_I$ 
4:      $L_{\text{align}} \leftarrow 0$ 
5:      $C_{DD} \leftarrow t_D(f_{I_p}(\text{aug}(B_D)))$ 
6:      $C_{ID} \leftarrow t_D(f_{I_p}(\text{aug}(B_I)))$ 
7:      $C_{DI} \leftarrow t_I(f_{I_p}(\text{aug}(B_D)))$ 
8:      $C_{II} \leftarrow t_I(f_{I_p}(\text{aug}(B_I)))$ 
9:      $D_{DD} \leftarrow \phi_D(f_D(\text{aug}(B_D)))$ 
10:     $D_{ID} \leftarrow \phi_D(f_D(\text{aug}(B_I)))$ 
11:     $D_{DI} \leftarrow \phi_I(f_D(\text{aug}(B_D)))$ 
         $\triangleright$  Compute cross-domain alignment losses
         $\mathcal{L}_{DD} = \mathcal{L}_{\text{CE}}(C_{DD}, D_{DD})$ 
         $\mathcal{L}_{II} = \mathcal{L}_{\text{CE}}(C_{II}, y_{I_i}^t)$ 
         $\mathcal{L}_{ID} = \mathcal{L}_{\text{CE}}(C_{ID}, D_{ID})$ 
         $\mathcal{L}_{DI} = \mathcal{L}_{\text{CE}}(C_{DI}, D_{DI})$ 
         $\triangleright$  Combine them
         $L_{\text{align}} += \mathcal{L}_{DD} + \lambda_1 \mathcal{L}_{II} + \lambda_2 \mathcal{L}_{ID} + \lambda_3 \mathcal{L}_{DI}$ 
12:   if  $t > 1$  then
13:      $\mathcal{L}_{\text{KD}} \leftarrow \text{KL}\left(\text{Softmax}\left(\frac{t_D^{t-1}}{\tau}\right) \parallel \text{Softmax}\left(\frac{t_D^t}{\tau}\right)\right) +$ 
         $\text{KL}\left(\text{Softmax}\left(\frac{t_I^{t-1}}{\tau}\right) \parallel \text{Softmax}\left(\frac{t_I^t}{\tau}\right)\right)$ 
14:      $\mathcal{L}_{\text{align}} += \lambda_{\text{KD}} \mathcal{L}_{\text{KD}}$ 
15:   end if
16:   Update  $f_{I_p}, t_D, t_I$  using  $\nabla L_{\text{align}}$ 
17: end for
18: return  $f_{I_p}, t_D, t_I$ 
19: end function

```

Algorithm 6 Stage 5: Replay Strategy

```

1: function BUILD REPLAY( $\mathcal{A}^t, f_{I_p}, t_D$ )
Require: Auxiliary pool  $\mathcal{A}^t$ ; prompted CLIP  $f_{I_p}$ ; down-
stream text embeddings  $t_D$ 
Ensure: Class balanced replay set  $\mathcal{A}_R^t$ 
2:   Initialize map  $\mathcal{G}[c] \leftarrow$  empty list for all classes  $c$ 
3:   for each image  $a \in \mathcal{A}^t$  do
4:      $v_p(a) \leftarrow \text{norm}(f_{I_p}(a))$ 
5:      $c^* \leftarrow \arg \max_c \cos(v_p(a), t_D[c])$ 
6:      $s(a) \leftarrow \max_c \cos(v_p(a), t_D[c])$ 
7:     append  $(a, s(a))$  to  $\mathcal{G}[c^*]$ 
8:   end for
9:    $\mathcal{A}_R^t \leftarrow \emptyset$ 
10:  for each class  $c$  do
11:    Sort  $\mathcal{G}[c]$  in descending order of  $s(a)$ 
12:    Add top  $k$  images in  $\mathcal{G}[c]$  to  $\mathcal{A}_R^t$ 
13:  end for
14:  return  $\mathcal{A}_R^t$ 
15: end function

```
