POSITION AWARE LAYER QUERIES FOR TEST TIME TRAINING IN VISION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-Time Training (TTT) adapts models to incoming test samples (*e.g.* out-of-distribution, (OOD)) when conventional fine-tuning is infeasible. Existing TTT methods for Vision-Language Models (VLMs) create supervision from several augmented views, *each* requiring forward (and often backward) passes through the entire VLM, incurring substantial computational cost. We observe that one forward pass with all the intermediate layer outputs already yields far more signal than the final embedding from all augmentations. We introduce **Layer Query Network (LQN)**, a lightweight approach that can adapt a frozen VLM (*teacher*) in a single forward pass of the VLM via a small model (*student*). LQN uses **Position-Aware Distillation (PAD)** to mimic the teacher VLM's intermediate-layer spatial tokens by *querying* spatial coordinates of intermediate tokens. LQN additionally relies on **Location Consistency Regularization (LCR)**, a self-supervision technique, replacing expensive $\mathcal{O}(H \times W)$ image augmentation with $\mathcal{O}(1)$ coordinate sampling. Integrating these, LQN i) Adapts and improves zero-shot CLIP ViT-B/16 by 9.8% Top-1 on OOD ImageNet, ii) Outperforms the previous best GS-Bias on fine-grained classification by 3.9% Top-1, iii) Faster convergence than TPS for CLIP ResNet-50 (47 mins vs 55 mins) iv) Generalizes adaptation to VLMs like SigLIP, EVA-CLIP, and CoCa, and lightweight students like MLP, ResNet, VGG, and v) Extends to panoptic, instance, and semantic segmentation.

1 INTRODUCTION

CLIP (Radford et al., 2021) is widely regarded as one of the first Vision-Language Models (VLMs) to demonstrate strong zero-shot generalization in tasks like image classification, image–text retrieval, image segmentation, video text retrieval, and audio classification (Wang et al., 2025b; Hur et al., 2025; Dixit et al., 2024). As real-world deployment of VLMs grows, a key question emerges: *"Is there an efficient way to boost the out-of-distribution generalization of VLMs for real-world?"*

VLMs degrade under distribution shift, *i.e.* unseen out-of-distribution (OOD) test domains (Shu et al., 2023). Parameter-efficient fine-tuning, such as Adapters (Yin et al., 2023), LoRA (Hu et al., 2022), VPT (Jia et al., 2022), NN & TK0 (Pathak et al., 2026), *etc*. can adapt VLMs but require labeled target data; an assumption that rarely holds in the real world. Improving this, *Test-Time Adaptation* 'peeks' at target data on-the-fly (Zhang et al., 2024), but relies on multiple test samples (*e.g.* cache seen samples) to refine predictions (Nguyen et al., 2025), limiting applicability in data-constrained scenarios *e.g.* medical diagnostics. Overcoming this constraint, Test-Time Training (**TTT**, Sun et al. (2020)) adapts using **just one test sample**. TTT frameworks like TPT (Shu et al., 2022) and TPS (Sui et al., 2025) adapt by self-supervising *consistency across multiple augmentations* for each test sample, (fig. 1 (a)). Although effective, this requires the VLM to perform multiple unnecessary forward (& backward) passes for each augmentation, incurring high computational cost.

We take a different view: the signal is already *inside* the VLM even without augmentations. A single forward pass produces not only a final embedding but also L layers of $H \times W$ spatial tokens of D-dimension (*e.g.* CLIP). We treat this stack as a *3D hidden-state volume* as *Teacher* output $Y_{img}^{1,2...L} \in \mathbb{R}^{H \times W \times L \times D}$ and fit a lightweight, randomly initialized small student network. Conditioned on the coordinates of the spatial tokens, this student mimics the teacher's feature at any spatial position and depth. Such coordinate-conditioned very small networks have the potential to fit smooth structure, avoiding noise (Rahaman et al., 2019; Ulyanov et al., 2018), thereby, potentially acting as a denoiser

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

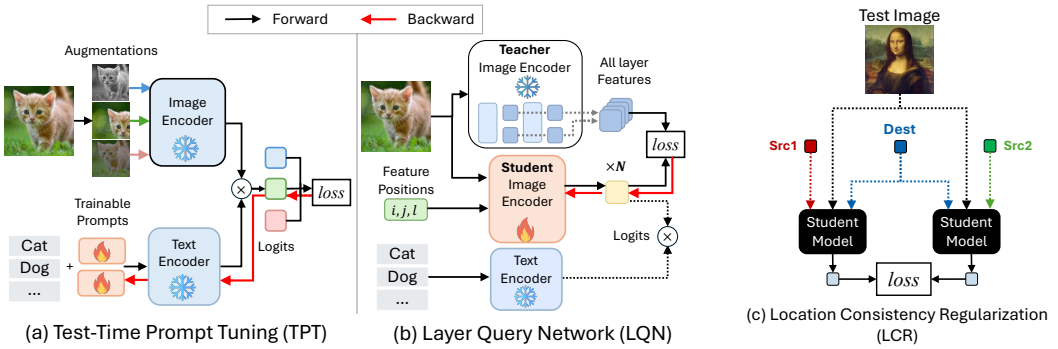


Figure 1: **Comparison with existing works:** *(a) Left:* TPT trains prompts via multiple augmentations, requiring multiple forward/backward passes of VLM *(b) Middle:* LQN trains small student in single forward pass from VLM (w/o any augmentation), by iteratively distilling intermediate features of image encoder via positional encoding (i, j, l coordinates for i, j spatial token at l layer) *(c) Right:* Self-supervision via varying src coordinates saves compute on expensive image augmentation and avoids over-reliance on VLMs (teacher) embedding on OOD samples.

for teacher's OOD features. We call this **Layer Query Network (LQN)**, a technique for adapting large frozen VLMs using a lightweight network in *only 1 VLM forward pass* per test sample.

LQN is composed of two complementary objectives: **Position-Aware Distillation (PAD)** and **Location Consistency Regularization (LCR)**. PAD distills the teacher VLM's token at randomly sampled coordinates ($dest$, (i, j, l)), encoded with a 3D RoPE (fig. 1 (b)). Unlike traditional layer-wise distillation (Romero et al., 2015), PAD does not need block-to-block correspondence between teacher and student, but the student can mimic every layer of a VLM. LCR is a self-supervised regularization on the student network without teacher supervision (fig. 1 (c)). By varying the input spatial coordinates src, the student maintains consistent predictions across locations. Sampling the src coordinate costs $\mathcal{O}(1)$, compared with expensive $\mathcal{O}(H \times W)$ image augmentations. Additionally, LCR reduces the *reliance* on the teacher for OOD inputs where VLM outputs may be unreliable.

LQN offers several benefits: **1) Efficient**: Distilling intermediate teacher layers helps small student networks converge in fewer iterations (reduced GFLOPs), surpassing existing TTT approaches. **(2) Teacher Adaptation:** LQN outperforms existing TTT approaches in adapting CLIP for ImageNet OOD variants and fine-grained benchmarks. **(3) Dense prediction:** Intermediate feature distillation helps the student learn fine-grained nuances of the teacher and improve dense prediction tasks. **(4) Self-Supervision**: Reduced reliance on the VLM, where teacher predictions may not be certain for OOD samples. **(5) Generalization** Since the input rely on coordinates, LQN can predict arbitrary intermediate features, making it easy to apply to a variety of VLMs and student networks.

In summary, Layer Query Network (**LQN**) is an efficient test-time training framework for *on-the-fly* adaptation of large VLMs using lightweight models without needing any image augmentations. LQN is evaluated on 14 classification benchmarks and 5 segmentation tasks. Key contributions include: (1) We introduce **PAD**, a position-aware distillation objective that fits all intermediate layers with one shared network, and **LCR**, a teacher-free anchor-consistency objective that replaces augmentation at $\mathcal{O}(1)$ cost. (2) On classification benchmarks, LQN improves CLIP ViT-B/16 and ViT-L/14 on ImageNet OOD variants by 9.8% and 5.7% over zero-shot, and outperforms the strongest TTT baselines by 3.9% on fine-grained classification benchmarks. (3) LQN extends to dense segmentation prediction by adapting EoMT on COCO, Cityscapes, and ADE20K (4) Generalization to VLMs like EVA, SigLIP, and CoCa, and lightweight students like ResNet & VGG.

2 Related Work

VLM Adaptation: VLMs are often adapted by fine-tuning the entire model or via LLM prompts (Pratt et al., 2023; Ren et al., 2023). Inspired by parameter-efficient approaches (Lester et al., 2021), adapter-based and cross-modal adaptation works have emerged (Gao et al., 2023; Lin et al., 2023); however, gradients still flow through the *entire* VLM, increasing optimization cost.

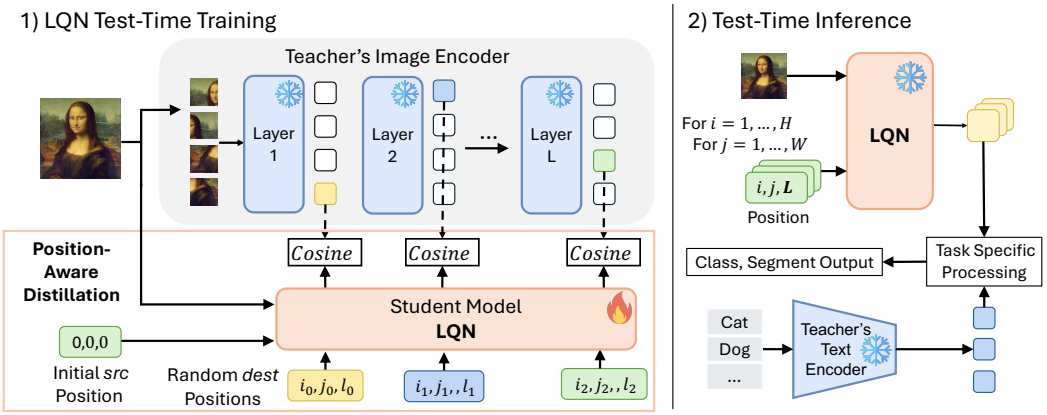


Figure 2: **Layer Query Network (LQN)** 1) For a single test image, *frozen* teacher (VLM) extracts intermediate features across all layers. During *test-time training*, LQN runs for N iterations by randomly sampling different destination *(dest)* positions (i, j, l), where (i, j) is the spatial location and l the layer depth. Position-Aware Distillation (**PAD**) takes the image, $(0, 0, 0)$ as the *src* position, and trains student network to predict the teacher's feature at *dest*. Location Consistency Regularization (**LCR**) self-supervises LQN to produce consistent features across random *src*(s). 2) After the final iteration (test-time inference), the student is queried at all spatial locations of the last layer L. Resultant features are processed by task-specific modules (eg, zero-shot classification/ segmentation).

Test-Time Optimization: Test-Time Adaptation (TTA) methods typically need *multiple* test samples *simultaneously* for stable adaptation (Prabhudesai et al., 2023; Yuan et al., 2023; Gong et al., 2022; 2024; Noori et al., 2025), while relying on historical information from multiple test samples. Test-Time Training (TTT) approaches like TPT (Shu et al., 2022) and prompt-tuning (Zhou et al., 2022b;a) enforce consistency across augmented views but require costly passes through the text encoder or require annotated training data. TPS (Sui et al., 2025) avoids this by adjusting precomputed feature vectors instead. MTA (Zanella & Ben Ayed, 2024), GS-Bias (Huang et al., 2025) further improve zero-shot robustness via mean-shift augmentation, and logit biasing, respectively. TTT-MAE(Gandelsman et al., 2022) uses SSL tasks like rotation/masked prediction, while Diff-TPT Feng et al. (2023) generates OOD data via Stable Diffusion. In contrast to APM (Modi & Rawat, 2024), we distill the teacher VLM's intermediate features into the student network, without relying solely on the teacher's features. Compared to previous works, LQN avoids dataset-specific pre-training, multiple expensive augmentations, and the VLM's separate forward / backward passes per augmentation, thereby saving compute.

3 Layer Query Network (LQN)

We introduce LQN, a **T**est-**T**ime **T**raining (TTT) framework designed to enhance the out-of-distribution generalization of Vision-Language Models (VLMs). First, we discuss the background on VLMs such as CLIP. Next, we discuss the **Position-Aware Distillation (PAD)** and **Location Consistency Regularization (LCR)**, which describes how LQN functions (section 3.2).

3.1 Preliminaries

A VLM like CLIP is typically pretrained on millions of image and text pairs. It consists of two parallel encoders, an image encoder $\mathcal{T}_{img}$ and a text encoder $\mathcal{T}_{text}$. Given an out-of-distribution test image x_{ood}, and a class label description cls, corresponding class probabilities are produced as:

$$Y_{\text{img}} = \mathcal{T}_{\text{img}}(x_{\text{ood}}), \quad Y_{\text{text}} = \mathcal{T}_{\text{text}}(\text{cls}), \quad \mathbb{P}_{\text{CLIP}}(y = c \mid (x_{\text{ood}}) = \frac{\exp(Y^c_{\text{text}} \cdot Y_{\text{img}}/t)}{\sum_{c'} \exp(Y^{c'}_{text} \cdot Y_{\text{img}}/t)} \quad (1)$$

where, Y_{img} and Y_{text} are vision and text embeddings, respectively, t is the temperature parameter in softmax and $\mathbb{P}_{\text{CLIP}}(y = c \mid x_{ood})$ is the normalized probability corresponding to this class label c. The max probability logit over C classes is chosen as the class prediction.

3.2 TEST-TIME TRAINING (TTT)

Following the standard TTT paradigm (Shu et al., 2022; Sui et al., 2025), our goal is to adapt a model (VLM) to an **unseen test sample** x_{ood} **within** N **iterations**. Conventionally, VLMs require multiple augmentations of x_{ood}, requiring multiple expensive forward/backward passes through the entire VLM. After adaptation, **weights are then reset before seeing the *next* test sample**, preventing information leakage across test samples. Our TTT framework adopts a teacher-student distillation framework (fig. 2, algorithm 1), where a lightweight trainable student (*e.g.* MLP) mimics a frozen VLM teacher (*e.g.* CLIP) for N iterations. Repeated forward / backward passes of *small* student are computationally much cheaper than those of the teacher VLM, requiring *only one* forward pass of the teacher (Y_{img}, and Y_{text}, eq. (1)). Student network is optimized to mimic the teacher's intermediate spatial tokens via **Position-Aware Distillation (PAD)**, and **Location Consistency Regularization (LCR)**. Finally, the student uses the teacher text embedding Y_{text} to generate the class logit.

3.2.1 LQN STUDENT NETWORK

Adapting / fine-tuning large-scale VLMs on *just one sample* risks overfitting and potentially losing VLM's zero-shot abilities (Pathak et al., 2026; Chen et al., 2024). Instead, we consider *adaptation by proxy*, where a small lightweight student can iteratively (N iterations) adapt on a single test sample x_{ood} (instead of large VLM). This student can 'learn' to mimic a larger VLM's internal embeddings.

Student network has three inputs: a test image x_{ood}, *src*, and *dest* coordinates of the internal spatial features of the VLM teacher. For example, 3D Coordinates like (4,5,2) represent the spatial coordinate $h, w = (4, 5)$, at the depth of 2^{nd} layer in the VLM teacher. *'Dest'* represents the location of the target spatial token in the teacher for distillation. The student is tasked to maintain a consistent output despite varying *input src* coordinates. This step may be considered equivalent to having multiple *input image augmentations*. While augmentation is computationally costly, O($H \times W$) pixels augmentation, sampling source coordinates is O(1) retrieval, thereby helping the model generate robust embedding efficiently. To *distinguish* between the locations of src and $dest$ (line 9 of algorithm 1), we concatenate $[\cdot|\cdot]$ an additional constant sinusoidal positional-encoding 'flag' $P^T \in \mathbb{R}^{2\times D}$. $P^T[0]$ and $P^T[1]$ indicates location of src and $dest$, respectively:

$$\text{src} = [\text{P}_{i_s,j_s,l_s}|\text{P}^\text{T}[0]], \quad \text{dest} = [\text{P}_{i_d,j_d,l_d}|\text{P}^\text{T}[1]] \tag{2}$$

3.2.2 POSITION-AWARE DISTILLATION (PAD)

Classically, knowledge distillation focuses on matching the final-layer features of the teacher (Shu et al., 2022; Wang et al., 2021). Additionally, distilling large networks (>200M params) to small networks (<30M params) often requires distilling intermediate layers (Hao et al., 2023; Yang et al., 2024), thereby improving dense-task predictions (*e.g.* segmentation). However, intermediate feature distillation often requires explicit correspondences between teacher and student blocks, *e.g.* block 1 features of the student mimic block 1 of the teacher; thereby constraining the student architecture.

To relax the above constraint, we propose *Position-Aware Distillation* (PAD). We collect intermediate spatial tokens across *all the layers* of VLM, $Y_{img}^{1,2,\dots L}$, where $Y_{img}^{l} \in \mathbb{R}^{H\times W\times D}$; L denotes all the layers of teacher VLM, and (H, W) are the all spatial tokens on layer l, with dimension of D. For the input test sample x_{ood}, the student S_{img} mimics the teacher's (i, j) spatial token on depth l, *i.e.* $S_{img}(x_{ood}, i, j, l) \rightarrow f$, where $f \in \mathbb{R}^D$ is the predicted spatial vector at position (i, j, l). For N iterations, the student randomly selects (uniform sampling) a batch of these spatial indices (i, j, l) and distills teacher embeddings on these indices. However, directly feeding 3D-position coordinates (i, j, l) to a neural network leads to poor convergence (Mildenhall et al., 2021). Therefore, we encode location (i, j, l) as a constant 3D RoPE positional-encoding $P(i, j, l)$ Su et al. (2023). The student takes as input a *fixed* source (src) location $(0, 0, 0)$ and a *random* destination $dest$ location (i_d, j_d, l_d), and learns to mimic teacher's token at $dest$ via Cosine loss (algorithm 1 (line 10)).

$$\text{L}_{\text{PAD}} = 1 - \left\langle \frac{S_{\text{img}}(\text{src}, \text{dest}, x_{\text{ood}}) \cdot \text{Y}_{\text{img}}[i_d, j_d, l_d]}{\|S_{\text{img}}(\text{src}, \text{dest}, x_{\text{ood}})\|_2 \|\text{Y}_{\text{img}}[i_d, j_d, l_d]\|_2} \right\rangle_{\text{cosine}} \tag{3}$$

Algorithm 1 LQN Pseudo-Code: Image classification shown, but can generalize for other tasks.

Input: OOD image x_{ood}, Class Labels cls(s)
Require: Teacher Image / Text Encoder $\mathcal{T}_{img}$ / $\mathcal{T}_{text}$, Student LQN $\mathcal{S}_{img}$, N iterations
1: $Y_{img}^{1,2...L} \leftarrow \mathcal{T}_{img}(x_{ood}) \in \mathbb{R}^{H\times W\times L\times D}$ // Teacher's all 'L' layers spatial image features
2: $Y_{text} \leftarrow \mathcal{T}_{text}(cls)$ // Teacher's text features
3: $\mathcal{S}_{img} \leftarrow N(\mu, \sigma)$ // Random initialization of student model before seeing x_{ood} test sample
4: **for** iteration k in N **do**
5: ******* **Position-Aware Distillation (PAD)**
6: $i_d, j_d, l_d \leftarrow$ Sample(H,W,L) // Uniform Sample 'dest' coordinates $\in [H, W, L]$
7: $P_{i_d,j_d,l_d} \leftarrow P(i_d, j_d, l_d)$ // RoPE Positional embedding of dest
8: dest $\leftarrow [P_{i_d,j_d,l_d} | P^T[1]]$ // Concat constant sinusoidal flag $P^T[1]$ to dest
9: src $\leftarrow [P_{0,0,0} | P^T[0]]$ // For const src $(0,0,0)$ concat constant sinusoidal flag $P^T[0]$
10: $L_{PAD} \leftarrow 1 - \left\langle \frac{\mathcal{S}_{img}(src,dest,x_{ood}) \cdot Y_{img}[i_d,j_d,l_d]}{\|\mathcal{S}_{img}(src,dest,x_{ood})\|_2 \|Y_{img}[i_d,j_d,l_d]\|_2} \right\rangle$ // $\mathcal{S}_{img}$ mimics teacher's 'dest' features
11: ******* **Location Consistency Regularization (LCR)**
12: (i_1, j_1, l_1) & $(i_2, j_2, l_2) \leftarrow$ Sample(H,W,L) // Sample two sources 'src1' & 'src2' like lines 6-9
13: src1 $\leftarrow [P_{i_1,j_1,l_1} | P^T[0]]$ & src2 $\leftarrow [P_{i_2,j_2,l_2} | P^T[0]]$ // concat flag $P^T[0]$ to src1 and src2
14: // $\mathcal{S}_{img}$ self-supervises to predict 'Dest' features from separate src1/src2
15: $L_{LCR} \leftarrow 1 - \left\langle \frac{\mathcal{S}_{img}(src1,dest,x_{ood}) \cdot \mathcal{S}_{img}(src2,dest,x_{ood})}{\|\mathcal{S}_{img}(src1,dest,x_{ood})\|_2 \|\mathcal{S}_{img}(src2,dest,x_{ood})\|_2} \right\rangle_{\cos} - \langle \log \|w_{src}\|_2 + \epsilon \rangle$
16: Loss $\leftarrow L_{PAD} + \alpha L_{LCR} \implies$ Update $\mathcal{S}_{img}$ // Update using both losses
17: **end for**
18: ******* **Prediction / Inference / Evaluation**
19: **for** $\forall (i, j) \in (H, W)$, last layer L **do**
20: $P_{i,j,L} \leftarrow P(i, j, L)$ Dest $\leftarrow [P_{i,j,L} | P^T[1]]$ & Src $\leftarrow [P_{0,0,0} | P^T[0]]$ // Constant source $(0,0,0)$
21: $Y_{img}^{Student}$ += $\mathcal{S}_{img}(src, dest, x_{ood})$ / $(H \cdot W)$ // Spatial averaged of x_{ood} features via S_{img}
22: **end for**
23: $P_{cls} \leftarrow Y_{img}^{Student} \cdot Y_{text}$ // Classification is product of student image features with teacher text features
24: **Output:** P_{cls}

3.2.3 Location Consistency Regularization (LCR)

Traditional TTT has shown the benefits of self-supervision consistency across multiple augmented views of the image. Instead of expensive $\mathcal{O}(H\times W)$ image augmentations, we vary input coordinates of the spatial tokens, *i.e.* sampling the src coordinates ($\mathcal{O}(1)$ cost). We name this *Location Consistency Regularization (LCR)*, a self-supervised objective that complements PAD. Intuitively, LCR encourages the student to produce consistent representations at a given *dest* across different *src(s)* locations, providing an additional signal beyond direct distillation (fig. 1 (c)). LCR is not an augmentation, but rather a coordinate-conditioned consistency constraint that serves as a proxy for augmentation consistency. In addition to serving as a cost-effective proxy to input augmentations, LCR reduces the blind reliance on teacher for OOD inputs where VLM outputs may be unreliable.

Here, we sample two random locations: $src1 = (i_1, j_1, l_1), src2 = (i_2, j_2, l_2)$ and a *single* destination location $dest = (i_d, j_d, l_d)$. The intuition is that, irrespective of whether the student operates on $(src1, dest)$ or $(src2, dest)$, it should predict the same representation at $dest$ every time

$$\mathrm{L_{LCR}} = 1 - \left\langle \frac{S_{\mathrm{img}}(\mathrm{src1}, \mathrm{dest}, x_{\mathrm{ood}}) \cdot \mathrm{S_{img}}(\mathrm{src2}, \mathrm{dest}, x_{\mathrm{ood}})}{\|S_{\mathrm{img}}(\mathrm{src1}, \mathrm{dest}, x_{\mathrm{ood}})\|_2 \|\mathrm{S_{img}}(\mathrm{src2}, \mathrm{dest}, x_{\mathrm{ood}})\|_2} \right\rangle_{\mathrm{cosine}} - \langle \log \|w_{\mathrm{src}}\|_2 + \epsilon \rangle \quad (4)$$

A trivial solution to the above is the model simply ignoring the source, *i.e.* weight of the source $w_{\mathrm{src}} \to 0$. We prevent this by adding an extra regularization (-log with $\epsilon = 1e-7$, clamped at max value of 0.7) on w_{src} to prevent weight collapse. While PAD *only* needs $dest$ ($src \leftarrow 0$), LCR uses randomly sampled src(s).

3.2.4 Inference and Loss

Both *Position-Aware Distillation (PAD)* and *Location Consistency Regularization (LCR)* are *jointly* performed on a test sample for N iterations. We further control the impact of LCR via α. Combining both (line 16 in algorithm 1), the final loss is given by:

$$\mathrm{Loss} = \mathrm{L_{PAD}} + \alpha \mathrm{L_{LCR}}. \quad (5)$$

Table 1: **Natural distribution shifts:** Top-1 Results for ImageNet and its distribution-shifted variants (ImageNet-A/-V2/-R/-Sketch). *Requirements* column specifies resources needed during test-time; *Augment.*: multiple augmented views of each test sample; LQN doesn't need any augmented views for test samples. Use of the CLIP variant as a teacher is shown on the left. Avg: across all cols, OOD Avg: all cols except ImageNet.

	Method	Augment.	ImageNet	ImageNet-A	ImageNet-V2	ImageNet-R	ImageNet-Sketch	Avg	OOD Avg
CLIP-B/16	CLIP-ViT-B/16	✗	66.7	47.8	60.8	73.9	46.0	59.0	57.1
	Ensemble	✗	68.3	49.8	61.8	77.6	48.2	61.1	59.4
	TPT	✓	68.9	54.7	63.4	77.0	47.9	62.4	60.8
	Diff-TPT	✓	70.3	55.6	65.1	75.0	46.8	62.6	60.6
	MTA + TPT	✓	70.0	58.0	64.2	78.3	49.6	64.0	62.5
	APM	✗	68.1	52.1	67.2	76.5	49.3	62.6	61.3
	GS-Bias	✓	70.5	56.6	64.6	80.4	50.3	64.5	63.0
	TPS	✓	70.1	60.0	64.7	80.2	49.9	65.0	63.7
	LQN *(Our)*	✗	73.1	60.8	71.2	83.1	52.6	68.2	66.9
	LQN (Pretrained, *Our*)	✗	74.5	62.3	72.9	84.8	53.9	69.7	68.5
CLIP-L/14	CLIP ViT-L/14	✗	76.2	69.6	72.1	85.9	58.8	72.5	71.6
	APM	✗	77.3	71.8	72.8	87.1	62.2	74.2	73.5
	LQN *(Our)*	✗	80.5	75.4	76.9	91.3	65.7	78.0	77.3
	LQN (Pretrained, *Our*)	✗	81.4	76.5	77.5	92.2	66.4	78.8	78.2

During evaluation, features corresponding to the *last-layer* L of the VLM teacher is predicted; *i.e.* $src = (0, 0, 0)$ and $dest = (i, j, L)$, for all $1 \leq i \leq H, 1 \leq j \leq W$, generating the final $Y_{img}^{student}$. Classification averages spatial features followed by the teacher's textual feature dot product (line 23 in algorithm 1). For image segmentation, $Y_{img}^{student}$ is directly fed to the teacher's frozen mask-head (frozen mask head + adapted LQN), where upsampled features produce the segmentation mask.

4 Experiments on LQN

Tasks and Datasets: LQN is evaluated on 14 classification benchmarks and 5 segmentation tasks. Following TPT (Shu et al., 2022), LQN is evaluated on two types of classification benchmarks: **1) Natural distribution shift**: evaluation on ImageNet val (2009), along with its distribution-shifted variants, namely ImageNet-A (2021b) , ImageNet-V2 (2019), ImageNet-R (2021a), and ImageNet-Sketch (2019). **(2) Cross-dataset generalization**: Experiments on 9 datasets, including Flowers102 (2008), DTD (2014), OxfordPets (2012), UCF101 (2012), Caltech101 (2022), Food101 (2014), SUN397 (2010), FGVCAircraft (2013), and EuroSAT (2019). **3) Dense Segmentation task:** Results on COCO (2014), ADE20K (2017), and Cityscapes (2016).

Evaluation Metrics For experiments with distribution shifts (Tab1, 4), we report Top-1 classification accuracy (across all datasets, namely ImageNet val, ImageNet-A, ImageNet-V2, ImageNet-R, ImageNet-Sketch), and Top-1 OOD Accuracy (all splits except ImageNet val). For dense segmentation tasks (Tab 3), we report Panoptic Quality (PQ), and Average Precision.

Implementation LQN student $(25M)$ contains 5 linear layers, with ReLU activation and one conv filter of 16×16 kernel, and stride 16. We evaluate two variants of LQN: LQN and LQN (Pretrained). (i) **LQN** uses PAD and LCR and is initialized from scratch. (ii) **LQN (Pretrained)** first performs SSL pre-training (distilling) ImageNet features from DINOv2, and then perform TTT on each sample independently. We *reset* the weights to pre-trained weights prior to processing a new sample. Detailed implementation details in Supplementary section B.

4.1 Main Results

1) Natural Distribution Shifts: Table 1 evaluates LQN on ImageNet and its 4 OOD variants. Zero-Shot CLIP underperforms, obtaining 57.1% average OOD accuracy. Compared with other TTT baselines (*e.g.* TPT 2022, Diff-TPT 2023, MTA + TPT, APM 2024, GS-Bias 2025, TPS 2025), LQN/LQN (pretrained) on average outperform by 4.9%/6.5% on OOD-Avg CLIP-ViT-Base. **2) Cross-Dataset Generalization:** Table 2 observes a similar performance improvement on fine-grained benchmarks. LQN/LQN (pretrained) on average outperforms previous results by 5.4%/6.8%, and even surpasses previous best GS-Bias. **3) Generalization to segmentation:** Table 3 shows the results for more challenging dense-level segmentation tasks, where LQN adapts

Table 2: **Cross-dataset generalization from ImageNet to fine-grained classification tasks.** Top-1 accuracy reported. CLIP-B/16 as teacher. Avg. averages all columns.

Method	Augment.	Flower102	DTD	Pets	UCF101	Caltech101	Food101	SUN397	Aircraft	EuroSAT	Avg
CLIP-ViT-B/16	✗	67.4	44.3	88.3	65.1	93.4	83.7	62.6	23.7	42.0	63.4
Ensemble	✗	67.0	45.0	86.9	65.2	93.6	82.9	65.6	23.2	50.4	64.4
TPT	✓	69.0	47.8	87.8	68.0	94.2	84.7	65.5	24.8	42.4	64.9
DiffTPT	✓	70.1	47.0	88.2	62.6	92.4	87.2	65.7	25.6	43.1	64.7
MTA	✓	68.0	45.9	88.2	68.6	94.2	85.0	66.6	25.2	45.3	65.2
APM	✗	62.0	48.9	81.6	72.6	89.6	84.2	65.7	29.7	55.7	65.5
GS-Bias	✓	71.9	46.1	90.3	67.5	94.6	86.0	67.4	26.4	52.4	67.0
LQN *(Our)*	✗	69.8	54.5	88.1	76.3	96.2	89.2	70.9	33.1	60.2	70.9
LQN (Pretrained, *Our*)	✗	71.1	55.9	89.4	77.5	97.1	90.8	72.4	34.2	61.8	72.2

Table 3: **Dense-task Segmentation** Input resolution of 1280×1280; semantic segmentation has 1024×1024. PQ: Panoptic Quality, AP: Average Precision. LQN performs TTT on EoMT (teacher).

Method	Panoptic		Instance	Semantic	
	COCO PQ ↑	ADE20K PQ ↑	COCO AP ↑	Cityscapes mIoU ↑	ADE20K mIoU ↑
EoMT	58.3	51.7	48.8	84.2	58.4
LQN *(Our)*	64.5	57.1	55.2	87.5	64.3
LQN (Pretrained, *Our*)	64.9	58.6	56.7	88.2	65.8

Table 4: **Generalization across various VLMs teacher** LQN+{SigLIP, EVA-CLIP for various classification benchmarks. Similar averaging as table 1 and table 2.

Method	ImageNet and OOD							Fine-grained classification								
	ImageNet	IN-A	IN-V2	IN-R	IN-Sketch	Avg	OOD Avg	Flowers102	DTD	Pets	UCF101	Caltech101	Food101	SUN397	Aircraft	EuroSAT	Avg
SigLIP	76.0	45.3	68.9	90.3	67.9	69.7	68.1	85.8	64.7	94.1	72.5	90.5	89.8	69.8	43.8	43.8	72.8
+ LQN	79.2	48.7	72.4	93.4	71.6	73.1	71.5	89.2	68.5	97.4	76.2	94.1	93.1	73.5	47.3	47.6	76.3
EVA-CLIP	76.1	64.6	68.9	89.1	63.3	72.4	71.5	72.0	59.3	93.7	74.1	90.4	89.7	71.9	28.5	69.9	72.2
+ LQN	79.8	68.9	73.1	92.5	67.4	76.3	75.5	76.4	63.2	97.2	78.6	94.3	93.7	75.9	32.7	74.1	76.2
CoCa	63.6	21.5	55.7	73.2	51.3	53.1	50.4	64.7	53.3	89.1	61.4	89.1	77.3	66.1	18.8	45.3	62.8
+ LQN	68.1	25.1	60.2	77.6	55.7	57.3	54.7	69.4	57.8	93.6	66.0	93.3	82.1	70.6	22.9	50.1	67.3

EoMT (Kerssies et al., 2025) as teacher. Figure 4 (left) shows a few sample qualitative segmentation results of LQN. **4) Generalization across VLMs:** Table 4 shows LQN's generalization across various VLMs other than CLIP like SigLIP (Zhai et al., 2023), EVA (Fang et al., 2023), and CoCa (Yu et al., 2022) 5) Extension to Test Time Augmentation (TTA): We have mainly focused on TTT; however, LQN can be generalized to TTA (Supplementary tab. 11), *i.e.* using multiple data points.

4.2 Analysis on LQN

FLOP Analysis: Figure 4 (middle) compares total GFLOPS consumed during TTT for both TPT and LQN. TPT only does one gradient update per test sample; however requires 63 augmentations, *i.e.* $63 + 1 = 64$ forward passes through VLM, totaling 1312 GFLOPS. In contrast, LQN performs one VLM forward pass (CLIP) costing 20.5 GFLOPs. However, the subsequent $N = 14$ backward steps for updating the student network cost 10 GFLOPs, yielding a total of $20.5 + 14 \times 10 = 160.5$ GFLOPs; far less than TPT. Figure 3 reports the *actual wall-clock time* (Zhang et al., 2024) required for TTT over 50,000 ImageNet validation images on a single A6000 GPU. APM consumes 1hr+, while LQN converges in 47mins; improving accuracy by 4.2%.

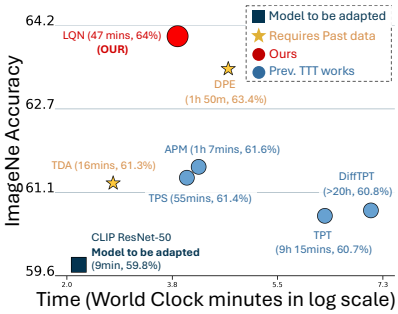


Figure 3: **Wall-clock time** CLIP ResNet-50 on ImageNet.

Unseen teacher layers: In Figure 4 (right), we use EoMT ViT-L as LQN's teacher for panoptic segmentation on COCO. When TTT is performed by distilling *all* 24 layers of the teacher into LQN, we obtain a peak mIoU of 64.5. Next, we perform TTT using only the *first 20 layers* of EoMT. However, during inference, we predict that the $20 - 24$ layer as input to the segmentation head, observing

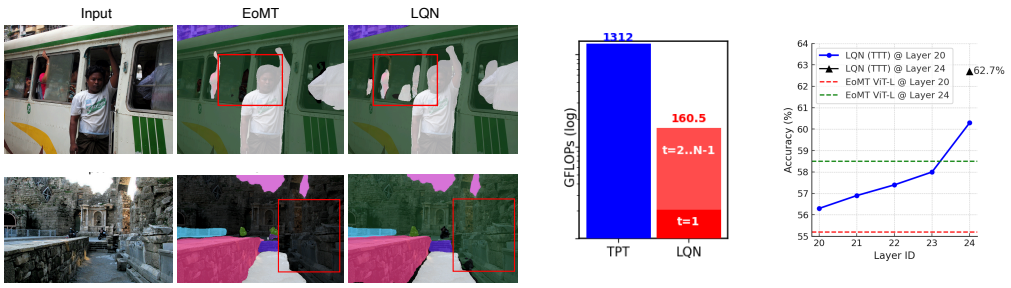


Figure 4: *Left:* **Segmentation Mask Quality: EoMT vs LQN** (Above) LQN can even segment persons *occluded* behind the window of the bus. (Bottom) LQN semantically groups visual elements of the scene (*e.g.* walls), whereas EoMT falls short. *Middle:* **Cost vs. Performance.** *Left)* TPT's 63 augmentations result in high overhead. *Right:* **Generalization:** Performance improves even at unseen final layers (COCO PQ), showing LQN's ability to generalize beyond TTT teacher layers.

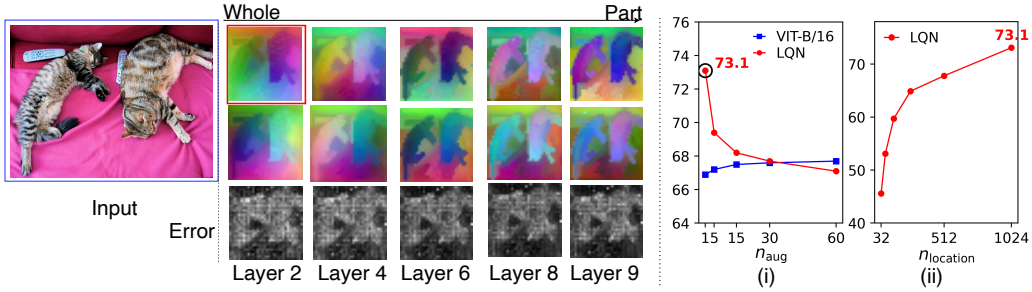


Figure 5: *Left:* Token visualizations of the teacher model EoMT (top), LQN (middle), and their difference (*Error*, bottom) showing LQN's predictions matching EoMT. *Right (i)* Augmenting test image during TTT *harms* LQN (red), but improves CLIP ViT-B/16 (blue). *(ii)* Distilling *more* teacher locations improves performance, verifying benefits of intermediate layer distillation. n_{aug} & $n_{location}$ represent the # of augmentations and the # of coordinates to distill in PAD.

a performance increase from 56.2 in the 20^{th} layer to 60.3 in the 24^{th} layer. This shows that the position-conditioned student can extrapolate to layers not seen during PAD.

LQN similarity with teacher's features: We feed-forward a sample COCO dataset image into both the EoMT teacher and LQN. Instead of predicting the final-layer features, we predict intermediate-layer spatial tokens. Fig 5 (left) shows the t-SNE of the intermediate representations of EoMT teacher and LQN prediction. We observe that the predicted features are very similar to the teacher's.

Why not augment test sample itself? In Fig. 5(right)(i), we perform augmentation of a same test-sample. This improves ViT-B based model consistent with prior observations in TPT/TPS. However, this harms LQN performance. Fig. 5(right)(ii), we gradually increase the number of locations (with uniform sampling) to distill over (PAD)[1]. We observe performance increases with more spatial co-ordinates, indicating the importance of distilling all locations.

Both src & dest vs just dest? Table 5 compares LQN variants with only $dest$ (no src) *i.e.* only PAD (LCR only makes sense with src). This has the worst performance, possibly because of *blind reliance* on the teacher for x_{ood}. Next, we apply LCR (eq. (4)) by enforcing self-consistency between locations $dest$ and $dest + \triangle$, where $\triangle$ is Gaussian noise. The intuition is that minor perturbations in dest should self-supervise and reduce reliance on the teacher (improved performance over only PAD). Finally, we test our LQN with consistency between $(src1, dest)$ & $(src2, dest)$, outperforming others. This indicates the importance of reduced reliance on the teacher, while maintaining self-consistency via *src*. We believe this is likely because src comes from a similar domain as other $dest$ input, while $\triangle$ Gaussian noise likely suffers from a domain mismatch with $dest$.

[1]Uniform sampling from a subset of $H \times W \times L$ teacher VLM internal locations (Algo 1 line 6).

Table 5: **LCR variations** W/o src, only PAD can be applied. $\triangle$: LCR w/ Gaussian noise instead of src.

Parts	Food-101	COCO
PAD (-src)	71.8	54.9
PAD + $\triangle$	76.0	58.1
PAD + LCR	89.2	64.5

Table 6: **Student Architectures**.

Student	Food-101
VGG	84.9
ResNet 18	85.4
ResNet 34	87.2
MLP	90.8

Table 7: Type of loss in PAD & LCR

Loss	Food-101	COCO
L1	75.8	52.0
MSE	85.7	61.1
Cosine	89.2	64.5

Table 8: **Positional Encodings**. Coord. feeds int positions.

Pos.	Food-101	COCO
Coord.	33.1	27.9
Sin-Cos	88.6	63.8
RoPE	89.2	64.5

Table 9: Sampling Strategies for VLM Layers.

Sampling Strategy	Food-101	COCO
Sampling more from shallow layers	81.9	43.7
Sampling more from deeper layers	87.8	52.0
Uniform Sampling	89.2	64.5

Table 10: Effect of Weight Regularization (Reg.) on LCR (Equation (4))

Configuration	Food-101	COCO
PAD + LCR (w/o weight Reg.)	87.4	62.6
PAD + LCR (w/ weight Reg.)	89.2	64.5

4.3 Ablations on LQN

We discuss various design choices made in LQN. We perform the following ablations on Food-101 classification (Top-1 Acc) and COCO segmentation (Panoptic Quality (PQ)).

LQN Student can work with multiple architectures: LQN can be generalized to standard small ConvNet networks like VGG/ResNet18/34 ($<$ 30 M params) as well. Table 6 shows the results for these backbones (ImageNet pretrained) as student networks. ResNets / VGG add an additional channel to the RGB input (3+1 channels) with the extra channel containing d-dimensional positional encoding repeated for all spatial locations. Using a trainable projector, 4D channel input is projected back to 3D input. All other layers are kept frozen.

Different Losses: Table 7 shows the effects of different losses applied to LQN for *both* PAD / LCR. Intuitively, L_1/L_2 constrain the student to exactly mimic the teacher's features ($>$ 200 M params), which may be harder given smaller student ($<$ 30 M params). We observe that cosine performs the best, since it enforces angular similarity between student and teacher (alignment).

Positional embeddings: Table 8 shows different strategies for injecting positional information in LQN. Simply injecting 3D integer coordinates harms performance. RoPE outperforms absolute encoding (Vaswani et al., 2017), indicating that rotating vectors in higher dimensions works well.

Sampling Strategy of teacher's locations: Table 9 shows results with different 'spatial token location' sampling strategies, *e.g.* uniform, higher probabilities of sampling from shallow / deeper layers. Uniform sampling appears to yield optimal results.

LCR Regularization: LQN's input consists of both src and $dest$. A trivial solution for LCR (Eq. 4) is when src weights $\rightarrow 0$, *i.e.* the output depends only on $dest$, ignoring src. Table 10 shows weight regularization $\langle \log \|w_{\text{src}}\|_2 + \epsilon \rangle$ has a visible performance impact in preventing this collapse.

5 Conclusion & Future Work

Our work shows it is possible to adapt VLMs on unseen test samples by updating a small proxy network, without relying on expensive augmentations or multiple forward/backward VLM passes. **L**ayer **Q**uery **N**etwork (LQN) allows us to mimic VLMs' intermediate features conditioned on 3D positional coordinates of the features (**P**ositional **A**ware **D**istillation, PAD), while reducing reliance on the teacher via **L**ocation **C**onsistency **R**egularization (LCR). Conditioning input on 3D coordinates helps generalize LQN across VLMs like CLIP, SigLIP, EVA, and CoCa, and smaller networks like MLP, ResNet, and VGG. Empirical generalization is shown across a variety of classification and segmentation tasks, while maintaining faster convergence compared to its counterparts.

A potential next step could be to generalize these capabilities for streaming videos, eg, adapting selectively/dynamically to incoming frames. In principle, memory/compute may be improved further by reducing the overhead of intermediate feature storage (during backprop), efficient feed-forward via top-k token selection, and feature compression.

Reproducibility Statement

To facilitate reproducibility, we have included the detailed implementation details and hyperparameters in the Supplementary. Supplementary section B describes the number of neurons in each layer of LQN. Similarly, Table 13 discusses the size of the hidden dimension, floating point precision, and hardware configuration of the computing machine on which experiments were performed.

AI Use Statement

In this work, we used generative AI tools for polishing our writing and drawing tables. We have not used generative AI tools for coding or performing experiments or for brainstorming the core technical contributions of this work. We have manually checked all the numbers in all the tables.

References

Yoshua Bengio, Geoffrey Hinton, Andrew Yao, Dawn Song, Pieter Abbeel, Trevor Darrell, Yuval Noah Harari, Ya-Qin Zhang, Lan Xue, Shai Shalev-Shwartz, et al. Managing extreme ai risks amid rapid progress. *Science*, 384(6698):842–845, 2024.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food101. In *European Conference on Computer Vision (ECCV)*, 2014.

Wei-Ting Chen, Yu-Jiet Vong, Sy-Yen Kuo, Sizhou Ma, and Jian Wang. Robustsam: Segment anything robustly on degraded images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4081–4091, 2024.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

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

Satvik Dixit, Laurie M Heller, and Chris Donahue. Vision language models are few-shot audio spectrogram classifiers. *arXiv preprint arXiv:2411.12058*, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 19358–19369. IEEE, 2023.

Chun-Mei Feng, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2704–2714, 2023.

Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei Efros. Test-time training with masked autoencoders. *Advances in Neural Information Processing Systems*, 35:29374–29385, 2022.

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*, pp. 1–15, 2023.

Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022.

Taesik Gong, Yewon Kim, Taeckyung Lee, Sorn Chottananurak, and Sung-Ju Lee. Sotta: Robust test-time adaptation on noisy data streams. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhiwei Hao, Jianyuan Guo, Kai Han, Yehui Tang, Han Hu, Yunhe Wang, and Chang Xu. One-for-all: Bridge the gap between heterogeneous architectures in knowledge distillation. *Advances in Neural Information Processing Systems*, 36:79570–79582, 2023.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8340–8349, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15262–15271, 2021b.

Geoffrey Hinton. How to represent part-whole hierarchies in a neural network. *Neural Computation*, 35(3):413–452, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Zhaohong Huang, Yuxin Zhang, Jingjing Xie, Fei Chao, and Rongrong Ji. Gs-bias: Global-spatial bias learner for single-image test-time adaptation of vision-language models. *arXiv preprint arXiv:2507.11969*, 2025.

Chan Hur, Jeong-hun Hong, Dong-hun Lee, Dabin Kang, Semin Myeong, Sang-hyo Park, and Hyeyoung Park. Narrating the video: Boosting text-video retrieval via comprehensive utilization of frame-level captions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24077–24086, 2025.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 2022.

Tommie Kerssies, Niccolò Cavagnero, Alexander Hermans, Narges Norouzi, Giuseppe Averta, Bastian Leibe, Gijs Dubbelman, and Daan de Geus. Your ViT is Secretly an Image Segmentation Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, Apr 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Zhiqiu Lin, Samuel Yu, Zhiyi Kuang, Deepak Pathak, and Deva Ramanan. Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19325–19337, 2023.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fgvc-aircraft. In *Technical report, Visual Geometry Group, University of Oxford*, 2013.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Rajat Modi and Yogesh Rawat. Asynchronous perception machine for efficient test time training. *Advances in Neural Information Processing Systems*, 37:106020–106058, 2024.

Khanh-Binh Nguyen, Phuoc-Nguyen Bui, Hyunseung Choo, and Duc Thanh Nguyen. Adaptive cache enhancement for test-time adaptation of vision-language models. *arXiv preprint arXiv:2508.07570*, 2025.

M-E Nilsback and A Zisserman. Flowers102. *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

Mehrdad Noori, David Osowiechi, Gustavo Adolfo Vargas Hakim, Ali Bahri, Moslem Yazdanpanah, Sahar Dastani, Farzad Beizaee, Ismail Ben Ayed, and Christian Desrosiers. Test-time adaptation of vision-language models for open-vocabulary semantic segmentation. *ArXiv*, abs/2505.21844, 2025. URL https://api.semanticscholar.org/CorpusID:278959297.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Priyank Pathak, Mukilan Karuppasamy, Aaditya Baranwal, Shruti Vyas, and Yogesh S. Rawat. Robust onion: Peeling open vocab object detectors under noise. In *Computer Vision – ECCV 2026*, pp. 353–373, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-37095-2.

Mihir Prabhudesai, Tsung-Wei Ke, Alexander Cong Li, Deepak Pathak, and Katerina Fragkiadaki. Diffusion-tta: Test-time adaptation of discriminative models via generative feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15691–15701, 2023.

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

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, pp. 5389–5400. PMLR, 2019.

Zhiyuan Ren, Yiyang Su, and xiaoming Liu. Chatgpt-powered hierarchical comparisons for image classification. *Advances in neural information processing systems*, 2023.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

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.

Yang Shu, Xingzhuo Guo, Jialong Wu, Ximei Wang, Jianmin Wang, and Mingsheng Long. Clipood: Generalizing clip to out-of-distributions. In *International conference on machine learning*, pp. 31716–31731. PMLR, 2023.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101. In *arXiv preprint arXiv:1212.0402*, 2012.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023.

Elaine Sui, Xiaohan Wang, and Serena Yeung-Levy. Just shift it: Test-time prototype shifting for zero-shot generalization with vision-language models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 825–835. IEEE, 2025.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Guo-Hua Wang, Yifan Ge, and Jianxin Wu. Distilling knowledge by mimicking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8183–8195, 2021.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in neural information processing systems*, 32, 2019.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025a.

Zhengyang Wang, Tingliang Feng, Fan Lyu, Fanhua Shang, Wei Feng, and Liang Wan. Dual semantic guidance for open vocabulary semantic segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20212–20222, 2025b.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun397. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

Zhendong Yang, Zhe Li, Ailing Zeng, Zexian Li, Chun Yuan, and Yu Li. Vitkd: Feature-based knowledge distillation for vision transformers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1379–1388, 2024. doi: 10.1109/CVPRW63382.2024.00145.

Dongshuo Yin, Leiyi Hu, Bin Li, and Youqun Zhang. Adapter is all you need for tuning visual tasks. *arXiv preprint arXiv:2311.15010*, 2023.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15922–15932, 2023.

Maxime Zanella and Ismail Ben Ayed. On the test-time zero-shot generalization of vision-language models: Do we really need prompt learning? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23783–23793, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Ce Zhang, Simon Stepputtis, Katia Sycara, and Yaqi Xie. Dual prototype evolving for test-time generalization of vision-language models. *Advances in Neural Information Processing Systems*, 37:32111–32136, 2024.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

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*, pp. 16816–16825, 2022a.

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, 2022b.