Distilling Vision Language Model for Robust Traffic Sign Perception in Autonomous Vehicles

Pedram MohajerAnsari, Amir Salarpour, Mert D. Pesé

***Abstract*— Traffic sign recognition (TSR) models based on deep neural networks achieve strong clean-data performance but remain vulnerable to physically realizable adversarial attacks, including shadow perturbations, natural-light interference, and printed patches. Existing defenses often improve robustness against one attack type while degrading performance on others, and can reduce clean accuracy. We propose LAMDA (Language-Anchored Model for Direction Alignment), a training framework that transfers language-grounded structure into TSR models without using adversarial examples or adding inference-time overhead. LAMDA builds two fixed prototype banks from VLM-generated sign descriptions and class names using a frozen OpenCLIP text encoder, and uses them to supervise visual features through two complementary auxiliary losses during training. At inference, the adapter and prototype banks are discarded, leaving a standard backbone and classifier. Evaluated on GTSRB and LISA across four backbones and three physical attack types, LAMDA is the only method among ten evaluated that consistently improves robustness across all attack–backbone–dataset combinations, with gains of up to $+12.5$ pp under shadow attacks and $+13.2$ pp under natural-light attacks, while preserving or improving clean accuracy in nearly all cases. Code is available at our repository: https://github.com/pedram-mohajer/LAMBDA.**

I. Introduction

Traffic sign recognition (TSR) is a fundamental component of modern autonomous systems, enabling vehicles to perceive and interpret traffic signs under diverse real-world conditions for safe driving operation [1]. Deep neural networks (DNNs) have become the standard solution for this task, with architectures such as ResNet [2], Swin Transformer [3], and ViT [4] achieving impressive accuracy on standard benchmarks. However, these models are vulnerable to adversarial examples (AEs), subtly modified inputs designed to fool DNNs [5], including physically realizable perturbations such as strategically placed shadows [6], natural-light interference [7], and printed patches placed directly on signs [8]. Such attacks pose a critical threat to TSR, where a single misclassification can lead to unsafe driving decisions [9].

Various defense methods have been proposed to mitigate such attacks: defensive distillation reduces model sensitivity but struggles to generalize across diverse perturbations [10]; input transformations can suppress adversarial noise but often degrade clean data quality, lowering accuracy [11]; and provable defenses, though theoretically robust, are computationally expensive and difficult to scale [12]. Among these, adversarial training [13] is the most widely adopted defense, yet it requires adversarial examples at training time and and trades off clean accuracy for robustness [14], [9].

We introduce **LAMDA** (**L**anguage-**A**nchored **M**odel for **D**irection **A**lignment), a training framework that improves TSR robustness against physical adversarial attacks without requiring any adversarial examples during training. The core idea is to use the rich semantic structure of language as a supervisory signal: before training begins, natural-language descriptions of each sign class are generated using NVILA [15] and, together with class names, are encoded by a frozen OpenCLIP text encoder [16] to form two fixed prototype banks. A lightweight adapter attached to the vision backbone projects image features into the same space as these prototypes during training, creating a channel through which language-grounded supervision reaches the visual representation. At inference, the adapter and prototype banks are discarded entirely, leaving a standard backbone and classifier with no added overhead.

LAMDA trains the backbone with two complementary auxiliary losses alongside standard cross-entropy. The first pulls image features toward the language description of the correct class, anchoring visual representations to the appearance-level semantics of each sign. The second uses the similarity between class names as soft training targets for the classifier head, providing a richer supervisory signal than one-hot labels. The two losses target distinct aspects of robustness and are stronger in combination than either is alone, trained entirely on clean data, with no modifications to the inference pipeline.

We evaluate LAMDA on GTSRB [17] and a 16-class LISA subset [18] across four backbones (ResNet-18, ResNet-34, Swin-T, and ViT-B/16) trained exclusively on clean data with no adversarial examples at any point during training. Adversarial examples are generated against two fixed target models, `gtsrb-cnn` and `lisa-cnn`, under three physically realizable attacks: shadow perturbations [6], natural-light interference [7], and the RP2 printable patch attack [8]. LAMDA is the only method that improves over the baseline on every attack, every backbone, and both datasets simultaneously, with gains of up to $+12.5$ pp under shadow attacks on GTSRB and $+13.2$ pp under natural-light attacks on LISA, while clean accuracy is preserved or

Pedram MohajerAnsari, Amir Salarpour, and Mert D. Pesé are with Clemson University, Clemson, SC, USA {pmohaje, asalarp, mpese}@clemson.edu

improved across all settings. In contrast, every one of the nine compared defenses degrades performance on at least one combination of backbone, attack, and dataset. The main contributions of this paper are:

- We propose LAMDA, a training framework that transfers language-grounded robustness into TSR models by aligning visual features with two fixed prototype banks, one built from VLM-generated sign descriptions and one from class names, through two complementary auxiliary losses. LAMDA requires no adversarial examples during training, adds no overhead at inference, and is the only method among ten evaluated that consistently improves robustness across all attacks, backbones, and datasets without degrading clean accuracy.
- We conduct a systematic evaluation on GTSRB and LISA across four backbones under three physically realizable attacks, comparing LAMDA against nine re-implemented defenses under identical training budgets. A comprehensive ablation over the two loss weights confirms that the two losses are complementary and super-additive, with a single weight configuration optimal across all evaluated conditions.

II. Related Work

Popp *et al.* [19] propose distilling a compact image encoder using supervision from a large CLIP-style teacher while keeping the text side fixed and class representations available as precomputed guidance. A key takeaway for our setting is the "text-off-device" deployment view: training can leverage rich language supervision (including synthetic prompts/data), but inference can rely on a lightweight vision backbone. This aligns with our goal of transferring VLM-derived robustness cues into an efficient TSR model without running a text encoder on-board. Li *et al.* [20] distill vision-language knowledge using prompts while explicitly reusing pre-stored class text features as supervision. Their pipeline precomputes text embeddings once (per class/prompt) and trains the student primarily through logit/feature alignment against these frozen targets, enabling inference with a compact image encoder plus a fixed text-prototype bank. This is highly similar in spirit to our frozen name/description prototype supervision, differing mainly in our task-specific TSR formulation and prototype construction.

Wu *et al.* [21] introduce a CLIP distillation framework that compresses CLIP by mimicking cross-modal affinities (image–text similarity structure) and using weight inheritance to stabilize training. TinyCLIP is a strong baseline for "CLIP-space" distillation because it preserves the pairwise alignment geometry that underlies zero-shot prompting. In contrast, our approach targets an image-only TSR student at inference time and uses frozen text prototypes as supervision rather than retaining a full student text encoder. Yang *et al.* [22] provide a systematic empirical study of CLIP distillation objectives, comparing feature matching, relational losses, contrastive objectives, and other KD variants across model scales. Their results clarify when simple feature/logit alignment is competitive and when additional relational constraints help, offering practical guidance for choosing robust and stable distillation signals. We leverage these insights to motivate our alignment losses against fixed language prototypes, while tailoring the objective to traffic-sign recognition and robustness.

Dong *et al.* [30] improve zero-shot robustness of VLMs by aligning representation subspaces induced by image augmentations and prompt variations (e.g., synonyms), and further aligning adversarial and clean subspaces via robust fine-tuning. This supports the broader thesis that language structure can act as an anchor for robust visual representations. Unlike their goal of robustifying the VLM itself for inference, we treat language features as frozen teachers and transfer robustness-relevant alignment into a compact TSR model that does not require a VLM at deployment.

TABLE I: Summary of baseline defenses compared against LAMDA, grouped by where they act in the pipeline. Each defense is applied on top of standard cross-entropy training under its best-performing hyperparameter configuration. **Note: `Adversarial training`** [13] is excluded as all methods are evaluated under a clean-training-only protocol.

Method	Category	Description
LS [23]	Train-time	*Label Smoothing*. Replaces one-hot labels with smoothed targets (e.g., 0.9 for the correct class, $0.1/(C-1)$ otherwise) to reduce overconfidence.
GT [24]	Train-time	*Gaussian Transformation*. Data augmentation with Gaussian blur/noise to encourage robustness to low-level distortions.
DO [25]	Train-time	*Dropout*. Randomly zeroes activations during training to reduce co-adaptation; acts as regularization.
JPG [11]	Preprocessing	*JPEG Compression*. Encodes/decodes images at fixed quality to suppress high-frequency artifacts (may remove useful detail).
BD [26]	Preprocessing	*Bit Depth Reduction*. Quantizes intensities to fewer levels (e.g., 3–5 bits/channel), attenuating fine perturbations.
MED [26]	Preprocessing	*Median Filtering*. Applies a 3×3 median filter to smooth pixel noise, often blurring edges.
HEQ [27]	Preprocessing	*Histogram Equalization*. Adjusts intensity distributions to enhance contrast; can alter colors.
RRP [28]	Evaluation-time	*Random Resized Cropping*. Randomly crops and resizes at test time, acting as a stochastic input transform.
RSE [29]	Evaluation-time	*Randomized Smoothing Ensemble*. Aggregates predictions over multiple noisy copies of the input to approximate a smoothed classifier.

III. LAMDA Framework

Motivated by the strong robustness we observe from VLMs on *unseen* adversarial inputs, our goal is to transfer part of this robustness into compact DNNs that can run in real time. To this end, we introduce **LAMDA** (Language-Anchored Model for Direction Alignment), a VLM-inspired training method that uses text prototypes to guide a standard vision backbone. During training, these prototypes provide an extra signal that nudges image features toward language-

TABLE II: Representative GTSRB/LISA sign images used to produce image embeddings via the backbone→adapter. Text prototypes (E_{desc}, E_{name}) are computed once with a frozen text encoder and kept fixed.

Dataset	Examples		
GTSRB	 White sign, red border, number 50; max speed 50 km/h.	 Red sign with white bar; no entry.	 Triangular sign, red border; pedestrian crossing.
LISA	 Octagonal red sign; mandatory stop.	 Yellow-green diamond; pedestrian crossing.	 Yellow diamond; signal ahead.

based directions. At inference, the network is a standard vision backbone and classifier with no VLM or text encoder in the loop, so latency and memory remain comparable to a conventional AV model.

LAMDA builds its text prototypes in two steps using off-the-shelf VLMs, as illustrated in Figure 2. First, for each traffic-sign image we query NVILA with the prompt: *"Describe the visual appearance of this traffic sign in one sentence, including its shape, colors, border, background, and any symbols, numbers, or text."* For each class we collect the resulting descriptions and also store a short class name such as "speed limit 50" or "no entry". Examples of image–description pairs are shown in Table II.

Second, we pass both the descriptions and the class names through a frozen OpenCLIP text encoder to obtain vector embeddings. We average and ℓ_2-normalize these embeddings to form two fixed prototype banks: one built from descriptions ($\mathbf{E}_{\text{desc}}$) and one from class names ($\mathbf{E}_{\text{name}}$). Both banks are computed once before training and never updated.

a) Formal definition.: Given an input x with label $y \in \{1, \dots, C\}$, define the backbone, head, and adapter mappings:

$$f_\theta : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^d, \quad h_W : \mathbb{R}^d \to \mathbb{R}^C, \quad g_\phi : \mathbb{R}^d \to \mathbb{R}^D.$$

The backbone produces a visual feature vector and the head maps it to class logits:

$$\mathbf{z} = f_\theta(x) \in \mathbb{R}^d, \qquad \mathbf{o} = h_W(\mathbf{z}) \in \mathbb{R}^C.$$

A lightweight adapter (two-layer MLP, hidden dim 512, batch norm, ReLU) projects $\mathbf{z}$ into the shared text space and is ℓ_2-normalized:

$$\hat{\mathbf{t}} = \text{norm}\big(g_\phi(\mathbf{z})\big) \in \mathbb{R}^D.$$

For each class c, descriptions are collected by prompting NVILA with representative class images, then deduplicated and templated (e.g., "a traffic sign: {description}"). Each prompt is encoded by the frozen text encoder, averaged, and normalized to form $\mathbf{e}_c^{\text{desc}} \in \mathbb{R}^D$. Likewise, class names

(a) Shadows [6]

(b) Natural Light [7]

(c) RP2 [8]

Fig. 1: Examples of physical perturbations. (a) *Shadows* [6]. (b) *Natural Light* [7]. (c) *RP2* [8].

are inserted into templates (e.g., "a traffic sign: {name}"), encoded, averaged, and normalized to form $\mathbf{e}_c^{\text{name}} \in \mathbb{R}^D$. Stacking across classes yields two fixed banks:

$$\mathbf{E}_{\text{desc}} \in \mathbb{R}^{C\times D}, \qquad \mathbf{E}_{\text{name}} \in \mathbb{R}^{C\times D}.$$

b) Alignment loss.: The adapter output is compared to description prototypes via cosine-similarity logits with temperature $\tau > 0$:

$$\mathbf{S}_{\text{align}}(x) = \frac{\hat{\mathbf{t}}\,\mathbf{E}_{\text{desc}}^\top}{\tau} \in \mathbb{R}^C.$$

An auxiliary cross-entropy treats $\mathbf{S}_{\text{align}}$ as logits for the target class, yielding the alignment loss:

$$\mathcal{L}_{\text{align}}(x, y) = \text{CE}\big(\mathbf{S}_{\text{align}}(x),\, y\big).$$

This term pulls $\hat{\mathbf{t}}$ toward the description prototype of the target class and away from all others, anchoring backbone features to semantic directions defined by natural language. The weight $\lambda \geq 0$, linearly warmed up in early epochs, controls the contribution of this term in the full objective.

c) Prototype loss.: The name prototypes induce a second regulariser at the head. For label y, retrieve its prototype $\mathbf{t}_y = \mathbf{e}_y^{\text{name}} \in \mathbb{R}^D$ and form soft targets by comparing against the full name bank with temperature $\tau_p > 0$ via the element-wise sigmoid σ:

$$\mathbf{q} = \sigma\big((\mathbf{t}_y\,\mathbf{E}_{\text{name}}^\top)/\tau_p\big) \in [0,1]^C.$$

Note that σ denotes the sigmoid (not softmax), so each entry of $\mathbf{q}$ is an independent soft target in $[0, 1]$, reflecting the degree of semantic similarity between the target class and every other class name. These soft targets are used in a

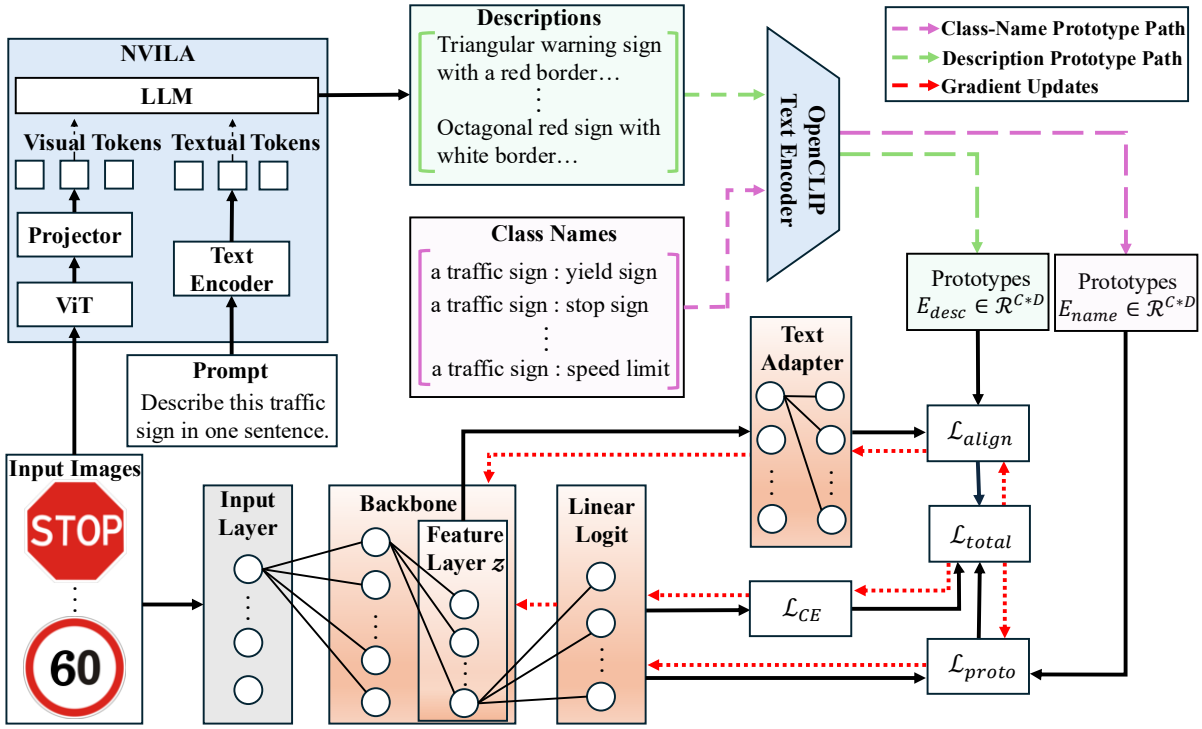


Fig. 2: **LAMDA (two-banks) training.** Trainable nodes (backbone, linear head, adapter) are updated during training; frozen nodes are fixed (the OpenCLIP text encoder and prototype banks). A single frozen OpenCLIP text encoder is shared by both parallel branches: *descriptions* are encoded once to build $\mathbf{E}_{\text{desc}}$ for adapter alignment via $\lambda\,\text{CE}(\mathbf{S}_{\text{align}}, y)$, where $\mathbf{S}_{\text{align}} = (\hat{\mathbf{t}}\,\mathbf{E}_{\text{desc}}^{\top})/\tau$; *class names* are encoded to $\mathbf{E}_{\text{name}}$, whose ground-truth row $\mathbf{t}_y$ induces soft targets $\sigma((\mathbf{t}_y\,\mathbf{E}_{\text{name}}^{\top})/\tau_p)$ for the head's prototype BCE weighted by μ. Total loss: $\mathcal{L} = \text{CE} + \lambda\,\text{CE}_{\text{align}} + \mu\,\text{BCE}_{\text{proto}}$.

prototype-based binary cross-entropy, where the head logits are scaled by a temperature $\tau_s > 0$ before comparison:

$$\mathcal{L}_{\text{proto}}(x, y) \;=\; \text{BCE}\big(\mathbf{o}/\tau_s,\; \mathbf{q}\big).$$

The weight $\mu \geq 0$ controls this term's contribution in the full objective. This loss encourages the head logits to reflect the semantic neighbourhood of the target class name, providing a smoother supervisory signal than one-hot labels.

d) Full objective.: The total training loss combines standard supervision with both auxiliary terms, each weighted by their respective scalars:

$$\boxed{\mathcal{L} \;=\; \mathcal{L}_{\text{CE}} \;+\; \lambda\,\mathcal{L}_{\text{align}} \;+\; \mu\,\mathcal{L}_{\text{proto}}}$$

where $\mathcal{L}_{\text{CE}} = \text{CE}(\mathbf{o}, y)$ is the standard classification loss on the head. Together, the two banks create complementary pressures: descriptions provide rich language grounding via $\mathcal{L}_{\text{align}}$, and class names enforce prototype-level regularisation via $\mathcal{L}_{\text{proto}}$. At inference, only the standard logits $\mathbf{o}$ are used for prediction; the text banks are not needed.

IV. Evaluation of LAMDA

We evaluate LAMDA on two public benchmarks for traffic-sign recognition: GTSRB with C=43 classes and LISA, where we follow the common 16-class subset. Training uses only the standard benign split for all methods; no adversarial examples are included during training. We study four backbones spanning CNNs and Transformers: ResNet-18, ResNet-34, Swin-T, and ViT-B/16. All methods share the same training schedule, augmentations, and data budget.

We evaluate against three physically realizable attacks, representative examples of which are shown in Figure 1. The shadow attack [6] synthesizes shadows by applying a polygonal mask over the sign region and attenuating the CIELAB L channel; mask parameters are optimized as a black-box attack using Expectation Over Transformations to preserve physical realizability. The natural-light attack [7] applies illumination via a parametric mask, modeling interference such as sunlight, headlights, and reflections, with parameters searched by zeroth-order optimization to maximize misclassification. The RP2 attack [8] optimizes a printable patch constrained to the sign surface using a victim dataset captured under varying viewing angles and distances, with synthetic transformations to simulate physical conditions. All adversarial examples are generated against fixed target models and kept unchanged for evaluation; methods that involve training are trained only on clean data and never see adversarial examples.

We compare LAMDA (λ=1, μ=1) against nine widely used defenses spanning three categories, summarized in Table I: train-time methods that modify the learning signal

TABLE III: GTSRB: Baseline accuracy and Δ accuracy (pp) vs. the CE-only baseline. ↑/↓ indicate improvement/degradation. LAMDA (λ=1, μ=1) is the only method that improves over the baseline on all four splits across all backbones.

Method	ResNet-18				ResNet-34				Swin-T				ViT-B/16			
	benign	light	shadow	RP2	benign	light	shadow	RP2	benign	light	shadow	RP2	benign	light	shadow	RP2
Baseline	96.010	75.770	61.379	50.43	96.078	73.833	56.663	55.27	99.477	75.625	71.065	66.91	98.981	71.121	68.110	58.91
$\lambda = 1$ $\mu = 1$	↑ **3.980**	↑ **3.641**	↑ **12.502**	↑ 4.351	↑ **3.912**	↑ **2.267**	↑ **9.278**	↑ **5.087**	↑ **0.513**	↑ **0.997**	↑ **3.322**	↑ 2.219	↑ **1.010**	↑ 4.290	↑ **8.231**	↑ 1.893
LS	↑ 0.203	↓ 3.709	↓ 3.709	↓ 2.670	↓ 3.564	↓ 4.881	↓ 7.079	↓ 0.520	↓ 2.469	↓ 3.758	↓ 11.710	↓ 8.680	↓ 4.491	↓ 11.011	↓ 18.711	↓ 4.190
GT	↓ 3.235	↓ 7.118	↓ 12.609	↓ 6.070	↓ 4.503	↑ 1.801	↓ 13.006	↓ 2.640	↓ 2.121	↑ 0.881	↓ 13.000	↓ 0.930	↓ 6.505	↓ 14.468	↓ 23.301	↓ 2.860
DO	↓ 2.169	↓ 1.133	↓ 8.261	↓ 7.330	↓ 3.109	↓ 0.010	↓ 6.856	↓ 0.960	↓ 1.850	↓ 3.283	↓ 10.064	↓ 1.480	↓ 5.198	↓ 18.245	↓ 20.154	↓ 1.950
JPG	↓ 1.811	↓ 4.639	↓ 11.786	↑ **8.740**	↓ 0.755	↓ 2.266	↓ 8.212	↑ 4.570	↓ 0.736	↓ 4.610	↓ 8.030	↓ 8.960	↑ **0.119**	↓ 2.315	↓ 0.785	↓ 0.770
BD	↓ 0.048	↓ 0.174	↓ 0.349	↓ 4.720	↓ 0.058	↑ **0.029**	↓ 0.232	↓ 2.730	↓ 0.048	↓ 0.320	↑ 2.274	↑ **4.200**	↑ 0.535	↓ 4.664	↓ 2.873	↑ **9.030**
MED	↓ 4.823	↓ 7.544	↓ 12.841	↓ 5.830	↓ 5.317	↓ 7.292	↓ 12.841	↓ 9.350	↓ 3.409	↓ 6.866	↓ 9.745	↑ 1.053	↓ 2.564	↓ 2.469	↓ 5.763	↑ 7.410
RRP	↓ 22.003	↓ 8.348	↓ 29.421	↑ 0.720	↓ 22.632	↓ 12.251	↓ 27.561	↓ 0.910	↓ 4.222	↓ 6.982	↓ 14.529	↓ 1.370	↓ 2.980	↓ 0.349	↓ 11.544	↓ 2.190
HEQ	↑ 0.001	↓ 0.001	↑ 0.001	↓ 0.050	↓ 0.001	↑ 0.001	↓ 0.001	↓ 0.100	↑ 0.001	↓ 0.001	↑ 3.000	↓ 0.020	↑ 0.555	↑ **5.249**	↑ 4.466	↓ 0.070
RSE	↓ 3.651	↓ 6.847	↓ 11.737	↑ 0.430	↓ 0.668	↓ 2.692	↓ 5.472	↓ 0.720	↓ 0.039	↓ 2.896	↓ 1.639	↓ 6.040	↓ 0.484	↑ 1.651	↑ 3.572	↓ 6.600

Per column, **bold** = largest improvement, underline = second largest among improving (green) cells.

or augmentations (label smoothing, Gaussian transformation, dropout); preprocessing methods that apply a fixed input transformation at inference (JPEG compression, bit-depth reduction, median filtering, histogram equalization, random resize-and-pad); and evaluation-time methods that change prediction via test-time stochastic transforms without updating parameters (randomized smoothing). Each defense is integrated into or on top of standard cross-entropy training (λ=0, μ=0), which serves as the reference baseline.

Tables III and IV report top-1 accuracy and Δ versus the baseline on GTSRB and LISA respectively; for each defense, we sweep its key hyperparameters and report the best-performing configuration. LAMDA is the only method that consistently improves over the baseline across adversarial splits, backbones, and datasets, while clean accuracy is preserved or improved in nearly all settings. (LISA ResNet-18: -0.23 pp). On GTSRB, gains are largest on shadow: $+12.50$ pp (ResNet-18), $+9.28$ pp (ResNet-34), $+8.23$ pp (ViT-B/16), and $+3.32$ pp (Swin-T). RP2 and AE-light also improve consistently across all four backbones, and benign accuracy is not degraded — it increases by $+3.98$ pp and $+3.91$ pp for ResNet-18 and ResNet-34, and by $+1.01$ pp and $+0.51$ pp for ViT-B/16 and Swin-T. On LISA, the dominant split shifts to AE-light: $+13.20$ pp (ResNet-34) and $+11.30$ pp (Swin-T), reflecting the lower baseline accuracy on light attacks in LISA (42.3% and 40.8% respectively) where the alignment signal provides the largest uplift. Shadow and RP2 also improve consistently across all backbones (shadow: $+10.15$ pp ViT-B/16, $+6.45$ pp ResNet-34; RP2: $+7.89$ pp ResNet-34, $+3.57$ pp ResNet-18).

Conventional defenses show inconsistent and often counterproductive effects across both datasets. Train-time methods generally degrade adversarial robustness despite being designed to improve generalization: on GTSRB, label smoothing and Gaussian transformation reduce shadow accuracy by up to -18.71 pp and -23.30 pp respectively for ViT-B/16, and on LISA their effects collapse entirely for ViT-B/16 (-37.83 pp and -23.18 pp on AE-light). Preprocessing defenses improve RP2 robustness in isolated cases (JPEG: $+8.74$ pp ResNet-18 on GTSRB; bit-depth: $+9.03$ pp ViT-B/16 on GTSRB) but consistently hurt shadow accuracy and, in the case of random resize-and-pad, severely degrade benign accuracy by up to -22.63 pp on GTSRB. Randomized smoothing causes catastrophic benign degradation on LISA for ResNet-18 (-71.76 pp), indicating instability on that distribution. In contrast, LAMDA achieves consistent improvements across all splits, backbones, and datasets without any benign accuracy tradeoff.

Across both datasets, CNN backbones (ResNet-18, ResNet-34) show larger absolute shadow gains on GTSRB owing to their lower shadow baselines (61.4% and 56.7% vs. 71.1% for Swin-T and 68.1% for ViT-B/16), while on LISA the transformer backbones show the largest AE-light improvements and CNNs lead on RP2. Across both datasets and all three attack types, no backbone is left unimproved by LAMDA on any adversarial split, confirming that the method generalises across both architecture families.

V. Ablation Study

We conduct a systematic ablation over the two auxiliary loss weights λ (alignment, $\mathcal{L}_{\text{align}}$) and μ (prototype, $\mathcal{L}_{\text{proto}}$), evaluating all combinations from $(\lambda, \mu) \in \{0, 1, 2\}^2 \backslash \{(0, 0)\}$ (eight configurations in total) against the CE-only $(0, 0)$ baseline. All other hyperparameters are held fixed throughout. We train and evaluate each configuration on both GTSRB and LISA across four backbones (ResNet-18, ResNet-34, Swin-T, ViT-B/16), yielding 72 trained models in total. Δ accuracy relative to $(0, 0)$ is reported across four splits (Benign, AE-light, AE-shadow, RP2) in Figures 3a and 3b, with absolute values for $(1, 1)$ in Tables III and IV.

Each loss component contributes independently. Comparing single-component configurations $(\lambda$=1, μ=0) and $(\lambda$=0, μ=1) reveals that the two losses target different aspects of robustness. On GTSRB, $(\lambda$=0, μ=1) yields stronger

TABLE IV: LISA: Baseline accuracy and Δ accuracy (pp) vs. the CE-only baseline. ↑/↓ indicate improvement/degradation. LAMDA (λ=1, μ=1) is the only method that improves over the baseline on all four splits across all backbones.

Method	ResNet-18				ResNet-34				Swin-T				ViT-B/16			
	benign	light	shadow	RP2	benign	light	shadow	RP2	benign	light	shadow	RP2	benign	light	shadow	RP2
Baseline	99.342	48.412	61.586	51.482	98.561	42.273	58.360	56.751	98.707	40.789	60.543	53.002	98.854	43.261	60.172	43.604
$\lambda = 1\ \mu = 1$	↓ 0.227	↑ **2.446**	↑ **4.704**	↑ **3.574**	↑ **0.876**	↑ **13.202**	↑ **6.452**	↑ **7.890**	↑ 0.854	↑ **11.296**	↑ **4.538**	↑ **2.084**	↑ 0.969	↑ 1.465	↑ **10.151**	↑ **2.847**
LS	↑ **0.512**	↑ 1.610	↑ 4.435	↓ 3.490	↓ 0.073	↓ 7.195	↑ 1.747	↓ 5.280	↓ 0.854	↑ 0.993	↓ 3.258	↓ 0.003	↓ 0.927	↓ 32.713	↓ 37.833	↓ 6.120
GT	↑ 0.439	↑ 1.570	↓ 0.806	↓ 3.670	↓ 0.146	↑ 8.963	↑ 0.780	↑ 4.750	↓ 2.054	↑ 0.164	↑ 0.462	↓ 0.440	↓ 0.219	↓ 21.593	↓ 23.183	↑ 0.710
DO	↑ 0.439	↓ 1.233	↑ 2.285	↓ 1.100	↓ 0.073	↓ 0.409	↑ 0.301	↓ 3.440	↓ 0.654	↓ 3.749	↑ 0.790	↓ 0.180	↓ 1.512	↓ 28.134	↓ 28.828	↑ 0.270
JPG	↓ 0.219	↑ 0.157	↓ 2.285	↓ 1.280	↓ 1.219	↓ 4.415	↓ 0.134	↓ 6.790	↑ **1.000**	↑ 1.000	↓ 1.226	↓ 0.270	↑ 0.927	↑ 1.711	↑ 7.194	↑ 0.180
BD	↓ 0.146	↑ 0.640	↓ 0.941	↑ 1.540	↓ 0.073	↓ 0.327	↑ 1.075	↑ 1.950	↓ 1.073	↓ 1.509	↑ 0.866	↓ 0.710	↑ **1.324**	↑ **3.673**	↑ 8.000	↓ 6.030
MED	↓ 0.951	↓ 4.177	↓ 9.274	↓ 2.700	↑ **0.342**	↓ 9.403	↓ 7.930	↓ 5.550	↓ 0.634	↓ 0.478	↓ 2.183	↓ 5.320	↓ 1.146	↓ 3.359	↑ 1.817	↓ 5.320
RRP	↓ 1.975	↓ 7.120	↓ 11.290	↓ 0.800	↓ 1.999	↓ 11.774	↓ 17.876	↓ 0.440	↑ 0.707	↓ 0.070	↓ 2.989	↓ 0.710	↑ 0.634	↓ 4.258	↓ 1.409	↓ 0.090
HEQ	↓ 0.805	↓ 1.315	↑ 0.538	↓ 0.090	↑ 0.561	↑ **3.271**	↑ 2.016	↑ **1.045**	↓ 0.746	↓ 0.308	↑ 2.925	↓ 0.090	↓ 0.073	↑ 1.164	↑ 1.253	↓ 0.090
RSE	↓ 71.763	↓ 25.681	↓ 60.753	↑ **0.180**	↓ 0.024	↓ 8.749	↓ 6.452	↑ 3.460	↓ 0.073	↑ 1.755	↓ 2.586	↓ 0.350	↓ 1.146	↑ 0.101	↑ 9.672	↓ 0.180

Per column, **bold** = largest improvement, underline = second largest among improving (green) cells.

AE-shadow gains than $(\lambda{=}1, \mu{=}0)$ across all backbones (+4.85 vs. +2.99 pp for ResNet-18; +3.52 vs. +2.50 pp for ResNet-34), while $(\lambda{=}1, \mu{=}0)$ provides more consistent improvement across all four splits simultaneously. This asymmetry is consistent across both datasets: $\mathcal{L}_{\text{proto}}$ drives the largest single-split gains while $\mathcal{L}_{\text{align}}$ contributes broader, more uniform improvement.

The two losses are complementary, not additive. The joint $(\lambda{=}1, \mu{=}1)$ configuration substantially exceeds both single-component settings on every split and backbone. On GTSRB ResNet-18, the AE-shadow gain at $(1, 1)$ is +12.50 pp — 2.6× the $(\lambda{=}0, \mu{=}1)$ gain of +4.85 pp and 4.2× the $(\lambda{=}1, \mu{=}0)$ gain of +2.99 pp — and this super-additive effect holds across all four backbones and both datasets without exception. Across all 72 runs, $(1, 1)$ achieves the highest accuracy on every evaluation split for every backbone on both datasets. On GTSRB, AE-shadow gains are +12.50 pp (ResNet-18, from a baseline of 61.38%), +9.28 pp (ResNet-34, baseline 56.66%), +8.23 pp (ViT-B/16, baseline 68.11%), and +3.32 pp (Swin-T, baseline 71.07%). AE-light and RP2 also improve consistently (RP2: +4.35 pp ResNet-18, +5.09 pp ResNet-34, +2.22 pp Swin-T, +1.89 pp ViT-B/16). Importantly, clean accuracy is not sacrificed: benign gains at $(1, 1)$ are +3.98 pp (ResNet-18) and +3.91 pp (ResNet-34), with smaller but still positive gains for the transformer backbones (+1.01 pp ViT-B/16, +0.51 pp Swin-T).

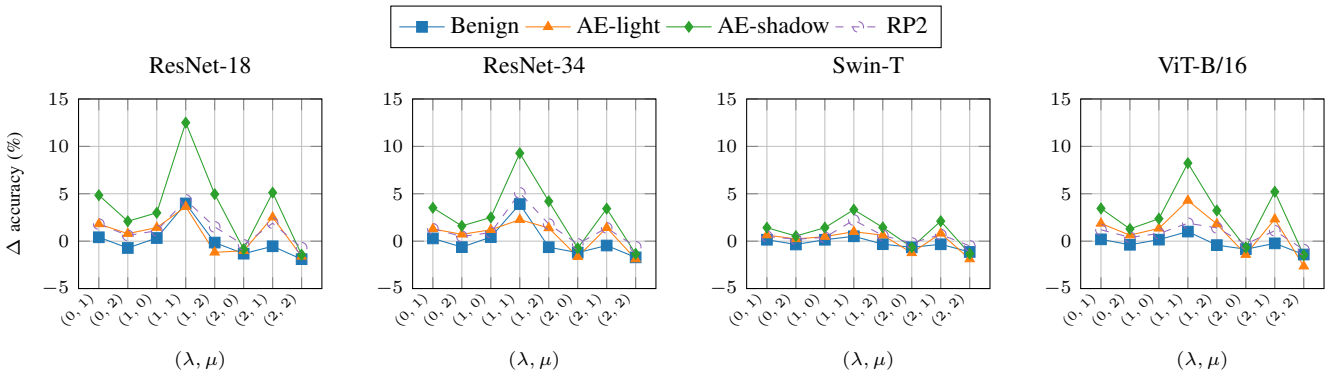
On LISA, the same $(1, 1)$ configuration is again optimal, but the dominant split shifts to AE-light: +13.20 pp (ResNet-34, from a baseline of 42.27%) and +11.30 pp (Swin-T, baseline 40.79%). AE-shadow and RP2 also improve across all backbones (e.g., shadow: +6.45 pp ResNet-34, +4.70 pp ResNet-18; RP2: +7.89 pp ResNet-34, +3.57 pp ResNet-18). The consistency of $(1, 1)$ as the optimum across two architecturally distinct datasets confirms that this weight setting generalises beyond the specific training distribution.

Gains do not scale monotonically with weight magnitude. Moving from $(1, 1)$ to $(2, 2)$ or from $(1, 0)$ to $(2, 0)$ consistently reduces performance, showing that increasing either weight beyond 1 is counterproductive. On GTSRB ResNet-18, $(\lambda{=}2, \mu{=}2)$ drops benign accuracy by −1.90 pp and AE-light by −1.61 pp relative to the CE-only baseline. This is not simply a case of high weights being uniformly bad, however: $(\lambda{=}2, \mu{=}1)$ retains positive AE-shadow gains on GTSRB (+5.12 pp ResNet-18, +3.43 pp ResNet-34, +5.21 pp ViT-B/16), showing that a moderate prototype weight can partially compensate for an overweighted alignment loss on shadow attacks specifically. The critical failure mode occurs when $\mu{=}2$ is combined with any non-zero λ: these configurations consistently degrade AE-light and benign accuracy across all backbones, suggesting that over-constraining the prototype space suppresses the representational flexibility needed for other attack types.

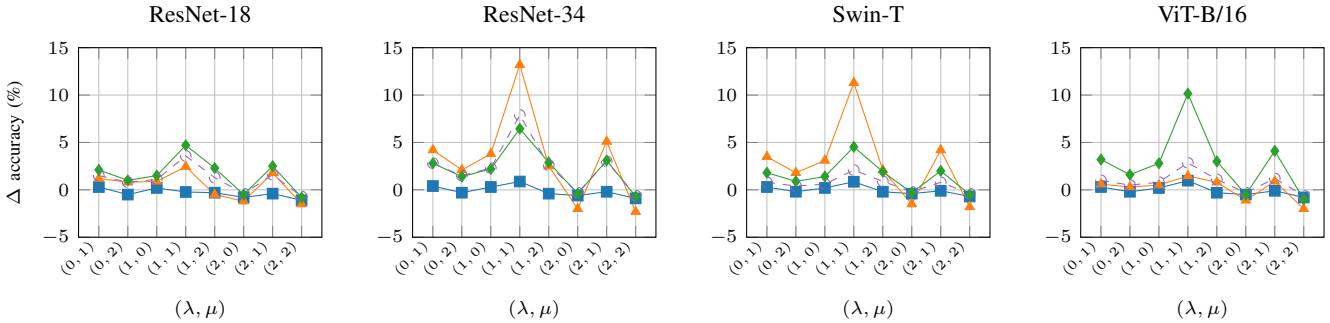
CNN and transformer backbones exhibit complementary sensitivity. On GTSRB, CNN backbones (ResNet-18, ResNet-34) show the largest absolute AE-shadow gains at $(1, 1)$, with ResNet-18 improving by +12.50 pp against a baseline of 61.38%. Transformer backbones start from stronger baselines on shadow (Swin-T: 71.07%, ViT-B/16: 68.11%) and correspondingly show smaller absolute gains (+3.32 pp and +8.23 pp respectively). On LISA the pattern inverts: transformers show the largest AE-light improvements (Swin-T: +11.30 pp, ViT-B/16: +1.47 pp), while CNNs dominate on RP2 and AE-shadow. Across both datasets, no backbone is left unimproved by $(1, 1)$ on any split, confirming that the method is architecture-agnostic.

A. Language vs. Random Prototypes

Replacing the language banks with irrelevant-content banks of identical dimensionality and retraining with $(\lambda{=}1, \mu{=}1)$ on GTSRB ResNet-18 (CE-only baseline: 96.01% benign, 61.38% shadow) reduces shadow accuracy by 6.41 pp while benign drops by only 2.46 pp. While, real descriptions yield +3.98 pp benign and +12.50 pp shadow.



(a) **GTSRB:** AE-shadow gains are most pronounced: +12.5 pp (ResNet-18), +9.3 pp (ResNet-34), +8.2 pp (ViT-B/16), +3.3 pp (Swin-T).



(b) **LISA:** The dominant split shifts to AE-light, with the largest gains for ResNet-34 (+13.2 pp) and Swin-T (+11.3 pp).

Fig. 3: Δ accuracy (pp) relative to the CE-only $(0, 0)$ baseline across eight (λ, μ) configurations for four backbones on GTSRB (a) and LISA (b). In both datasets, $(1, 1)$ is the global optimum across all splits and backbones, while high-weight configurations consistently degrade performance.

Model	Setting	76 cm	203 cm	292 cm	458 cm
ResNet-18	Baseline	✓ SL25 (0.38)	✗ SigAh (0.40)	✗ Sch (0.83)	✗ Sch (0.72)
	λ=1, μ=1	✓ SL25 (0.64)	✓ SL25 (0.48)	✓ SL25 (0.35)	✗ TrnR (0.23)
ResNet-34	Baseline	✓ SL25 (0.42)	✗ SigAh (0.45)	✗ SigAh (0.95)	✗ Sch (0.95)
	λ=1, μ=1	✓ SL25 (0.47)	✓ SL25 (0.35)	✓ SL25 (0.33)	✓ SL25 (0.23)
Swin-T	Baseline	✓ SL25 (0.62)	✓ SL25 (0.41)	✗ SigAh (0.37)	✗ Sch (0.32)
	λ=1, μ=1	✓ SL25 (0.59)	✓ SL25 (0.40)	✗ Sch (0.35)	✗ Sch (0.31)
ViT-B/16	Baseline	✓ SL25 (0.56)	✓ SL25 (0.50)	✗ Mrg (0.29)	✗ StpAh (0.31)
	λ=1, μ=1	✓ SL25 (0.43)	✓ SL25 (0.46)	✓ SL25 (0.33)	✗ Yld (0.27)

✓ = correct (Speed Limit 25), ✗ = misclassified (attack succeeded). Confidence in parentheses.

Fig. 4: Physical adversarial patch attack (RP2) on a *Speed Limit 25* sign evaluated at four distances. **Baseline**: standard training (λ=0, μ=0). Our method (λ=1, μ=1) improves robustness, correctly classifying the attacked sign at distances where the baseline fails.

VI. Real-World Experiments

We conduct a physical-world experiment using the RP2 adversarial patch attack [8]. We generate printable adversarial patches targeting a *Speed Limit 25* sign. The patches consist of two horizontal bars (covering approximately 19% of the sign surface) printed on paper and physically attached to a real traffic sign. We then capture photographs of the attacked sign at four increasing distances: 76 cm, 203 cm, 292 cm, and 458 cm. Each image is classified by four backbone architectures (ResNet-18, ResNet-34, Swin-T, and ViT-B/16) under both the baseline and our proposed method (λ=1, μ=1). As shown in Figure 4, the baseline models are highly vulnerable to the physical attack, correctly classifying the sign in only 6 out of 16 cases (37.5%). In contrast, *LAMBDA* correctly identifies the sign in 11 out of 16 cases (68.8%), representing a significant improvement of over 31 percentage points. Notably, ResNet-34 with our defense achieves 100% accuracy across all distances.

VII. Future work

Physical attacks in our evaluation are generated against fixed target models rather than the deployed network, an inherently transfer-based setting. Extending the analysis to adaptive gradient-based attackers with direct access to the backbone is left to future work.

VIII. Conclusion

We presented LAMDA, a training framework that improves the robustness of traffic sign recognition models by anchoring visual representations to frozen language prototypes derived from an off-the-shelf VLM. During training, LAMDA uses two offline prototype banks and two complementary auxiliary losses, requiring no adversarial examples and incurring no inference-time overhead. Across GTSRB and LISA, four backbones, and three physically realizable attacks, LAMDA is the only evaluated method that consistently improves adversarial robustness across all attack–backbone–dataset combinations, while preserving or improving clean accuracy in nearly all cases. In particular, LAMDA achieves gains of up to $+12.5$ pp under shadow attacks on GTSRB and $+13.2$ pp under natural-light attacks on LISA, and improves physical RP2 robustness from 37.5% to 68.8% in our real-world experiment. Ablations further show that the alignment and prototype losses are complementary, with a single $(\lambda{=}1, \mu{=}1)$ configuration generalizing across datasets and architectures. These results suggest that language-grounded supervision offers a practical path toward robust TSR without adversarial data, deployment overhead, or substantial clean-accuracy tradeoffs.

Acknowledgments

This work was supported in part by a grant from The BMW Group.

References

[1] J. S. O. Medina, J. G. M. Lázaro, A. Rassõlkin, and M. Ibrahim, "The road ahead: A comprehensive review of recent advances in traffic sign and lane line recognition for autonomous systems," *IEEE Open Journal of Vehicular Technology*, 2025.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[6] Y. Zhong, X. Liu, D. Zhai, J. Jiang, and X. Ji, "Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 345–15 354.

[7] T.-F. Hsiao, B.-L. Huang, Z.-X. Ni, Y.-T. Lin, H.-H. Shuai, Y.-H. Li, and W.-H. Cheng, "Natural light can also be dangerous: Traffic sign misinterpretation under adversarial natural light attacks," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 3903–3912.

[8] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1625–1634.

[9] P. MohajerAnsari, A. Salarpour, M. Kühr, S. Huang, M. Hamad, S. Steinhorst, H. Olufowobi, and M. D. Pesé, "On the natural robustness of vision-language models against visual perception attacks in autonomous driving," *arXiv preprint arXiv:2506.11472*, 2025.

[10] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 582–597.

[11] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten, "Countering adversarial images using input transformations," *arXiv preprint arXiv:1711.00117*, 2017.

[12] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International conference on machine learning*. PMLR, 2018, pp. 5286–5295.

[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[14] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," *arXiv preprint arXiv:1805.12152*, 2018.

[15] Z. Liu, L. Zhu, B. Shi, Z. Zhang, Y. Lou, S. Yang, H. Xi, S. Cao, Y. Gu, D. Li *et al.*, "Nvila: Efficient frontier visual language models," *arXiv preprint arXiv:2412.04468*, 2024.

[16] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, "Reproducible scaling laws for contrastive language-image learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 2818–2829.

[17] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *IEEE International Joint Conference on Neural Networks*, 2011, pp. 1453–1460.

[18] Georgia Tech Digital Intelligence Systems Laboratory (DiSL). (2025) LISA traffic sign dataset. GTDLBench. Accessed: 2025-09-24. [Online]. Available: https://git-disl.github.io/GTDLBench/datasets/lisa_traffic_sign_dataset/

[19] N. Popp, J. H. Metzen, and M. Hein, "Zero-shot distillation for image encoders: how to make effective use of synthetic data," *arXiv preprint arXiv:2404.16637*, 2024.

[20] Z. Li, X. Li, X. Fu, X. Zhang, W. Wang, S. Chen, and J. Yang, "Promptkd: Unsupervised prompt distillation for vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 617–26 626.

[21] K. Wu, H. Peng, Z. Zhou, B. Xiao, M. Liu, L. Yuan, H. Xuan, M. Valenzuela, X. S. Chen, X. Wang *et al.*, "Tinyclip: Clip distillation via affinity mimicking and weight inheritance," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 21 970–21 980.

[22] C. Yang, Z. An, L. Huang, J. Bi, X. Yu, H. Yang, B. Diao, and Y. Xu, "Clip-kd: An empirical study of clip model distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 952–15 962.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[24] C. M. Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural computation*, vol. 7, no. 1, pp. 108–116, 1995.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

[27] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics gems IV*, 1994, pp. 474–485.

[28] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," *arXiv preprint arXiv:1711.01991*, 2017.

[29] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *international conference on machine learning*. PMLR, 2019, pp. 1310–1320.

[30] J. Dong, P. Koniusz, L. Feng, Y. Zhang, H. Zhu, W. Liu, X. Qu, and Y.-S. Ong, "Robustifying zero-shot vision language models by subspaces alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 21 037–21 047.