ORION: ORthonormal Text Encoding for Universal VLM AdaptatION

Supplementary Material

A. Datasets and Prompt Templates

Following the standard practice in the literature [27, 37, 42], we evaluate on 11 widely used recognition benchmarks covering a broad range of domains, label granularities and visual statistics. Fine-grained object datasets include OxfordPets ("Pets") [25], StanfordCars ("Cars") [17] and FGVCAircraft ("Aircraft"); texture and material recognition is evaluated on Describable Textures (DTD) [7] and Food101 ("Food") [3]; scene and remote–sensing understanding is assessed on SUN397 ("SUN") [34] and EuroSAT [11]; and generic visual recognition is covered by Flowers102 ("Flowers") [22], Caltech101 ("Caltech") [10], UCF101 ("UCF") [30] and ImageNet [9]. For all datasets we follow the official or commonly adopted train/test splits and evaluation protocols used in CLIP and CoOp works. We summarize the basic statistics and task descriptions for each benchmark in Table 7. Table 8 lists the three prompt templates used to train the text encoder with class names for each dataset. These templates are used *only* for **ORION** pretraining stage; all baselines are evaluated with their respective default prompts. Together, these details are intended to facilitate faithful reproduction of our experiments.

Table 7. Additional information on the evaluation datasets. We follow the standard class splits and test protocols used in CLIP/CoOp works.

Dataset	Other name	#classes	#test samples
SUN397	SUN397	397	19,850
FGVCAircraft	Aircraft	100	3,333
EuroSAT	EuroSAT	10	8,100
StanfordCars	Cars	196	8,041
Food101	Food101	101	30,300
OxfordPets	Pets	37	3,669
Flowers102	Flowers102	102	2,463
Caltech101	Caltech101	101	2,465
DTD	DTD	47	1,692
UCF101	UCF101	101	3,783
ImageNet	ImageNet	1000	50,000

B. Additional Benchmark Protocols

Zero-shot evaluation. For zero-shot evaluation, we replace the original CLIP/MetaCLIP text encoder by our orthogonal encoder and keep the vision backbone frozen. All methods are evaluated on the same 11 benchmarks described in the main paper (Pets, Cars, Aircraft, DTD, Food101, SUN397, EuroSAT, Flowers102, Caltech101, UCF101, ImageNet) using the standard class splits and test protocols from prior CLIP/CoOp works [27, 41, 42].

Table 8. Prompt templates used for training the text encoder on each dataset. Curly braces {} indicate the position where the class name is inserted.

Dataset	Prompt templates
ImageNet	"a photo of a {}", "a picture of a {}", "an image of a {}"
SUN397	"a photo of a {}", "an indoor scene of {}", "an outdoor scene of {}"
FGVCAircraft	"a photo of a {}, a type of aircraft", "an in-flight {} aircraft", "a parked {} airplane"
EuroSAT	"a satellite photo of {}", "an aerial photo of {}", "a high-resolution satellite image of {}"
StanfordCars	"a photo of a {} car", "a showroom photo of a {}", "a street photo of {}"
Food101	"a photo of {}", "a plated dish of {}", "a close-up of {}"
OxfordPets	"a photo of a {}", "a portrait of a {}", "a close-up photo of a {}"
Flowers102	"a close-up photo of a {} flower", "a macro photograph of {}", "a garden photograph of {}"
Caltech101	"a photo of a {}", "a centered photo of {}", "a studio photo of {}"
DTD	"{} texture", "a close-up of {} pattern", "a macro shot showing {} texture"
UCF101	"a photo of a person doing {}", "an image of someone performing {}", "a video frame of {}"

Few-shot protocols (CoOp and CLAP). In the few-shot setting, we follow the standard CoOp and CLAP training protocols. For each dataset and shot value K, we randomly sample K labeled examples per class on the training split and train CoOp [42] (context-based prompt learning) or CLAP [29] (adapter-augmented tuning) on these support sets. Our method is integrated by initializing all textual prototypes in CoOp/CLAP with the orthogonal embeddings produced by **ORION**, while keeping the vision backbone and all other hyperparameters identical to the original implementations. We report the mean accuracy over five random K-shot splits for each dataset.

Test-Time Adaptation: MTA and TPT. For the TTA regime, we adopt the original protocols of MTA and TPT.
MTA. MTA [36] models the class scores as a balanced mixture of Gaussians in the CLIP embedding space and performs one-shot transductive adaptation on the entire test set, assuming all classes are present in the batch. We plug our orthogonal prototypes into the MTA initialization and keep all optimization hyperparameters (such as, number of EM iterations, batch size, temperature) fixed to those used in the official implementation.
TPT. For TPT [28], we perform test-time prompt tuning on each individual test image. Following the original paper, we generate multiple augmented views of the image, minimize the marginal entropy across views and use confidence selection to discard high-entropy (low-confidence) augmenta-

tions. In our integration, only the initial textual prototypes are changed to **ORION** embeddings; the CLIP backbone, augmentation pipeline, number of views, and optimization schedule are kept identical to the original TPT setup.

Realistic test-time adaptation: StatA. For StatA, we strictly follow the realistic batch and online TTA scenarios introduced in [38]. **Batch-realistic setting.** Each task corresponds to a batch of test samples with a limited number of effective classes K_{eff}. We consider the six ranges defined in StatA: Very Low (1–4 classes), Low (2–10), Medium (5–25), High (25–50), Very High (50–100), and All (all classes present). For each dataset and scenario, 1,000 tasks are generated and accuracy is averaged over tasks, exactly as in [38]. **Online-realistic setting.** In the streaming scenario, test data arrive as a sequence of correlated batches. Following StatA, we control the temporal correlation between batches using a Dirichlet distribution with concentration parameter γ over class proportions: low, medium, and high correlation correspond to $\gamma \in \{0.1, 0.01, 0.001\}$, and the *Separate* setting feeds classes sequentially [38]. We adopt the same number of tasks (100 streams per configuration), batch size, and optimization schedule as StatA. In both batch-realistic and online-realistic regimes, we simply replace CLIP's original textual prototypes by **ORION** and leave all StatA hyperparameters unchanged, isolating the effect of our orthogonal text encoder on robustness across varying K_{eff} and Dirichlet-controlled correlations.

B.1. Extended Analysis of SVD Orthogonalization (MTA)

We expand the Table 6 of the main paper, in Table 9, to report the per-dataset comparison of the training-free SVD variant against both the MTA baseline and our learnable **ORION** encoder. Across almost all datasets, we see that enforcing hard orthogonality in closed form via SVD degrades performance, while **ORION** yields systematic gains.

Compared to the baseline MTA, SVD whitening reduces accuracy on 10 out of 11 datasets, with drops of 4–5 points on EuroSAT ($45.36\% \rightarrow 40.90\%$) and ImageNet ($69.11\% \rightarrow 64.20\%$), and over 11% on Flowers ($68.26\% \rightarrow 57.10\%$). Even on datasets where the drop is smaller (e.g., DTD, UCF101), SVD never recovers the original baseline performance. This confirms that globally rotating the prototype matrix to an exactly orthogonal basis removes useful semantic anisotropy: classes that should remain moderately related (e.g., visually similar object or scene categories) are forced to be equally distant, which disrupts the alignment between visual and textual features learned during pre-training.

In contrast, **ORION** improves over MTA on every dataset, with average accuracy increasing from 65.87% to 67.53%. The gains are especially pronounced on challenging, fine-grained or texture-centric benchmarks such as Aircraft ($25.32\% \rightarrow 27.00\%$), EuroSAT ($45.36\% \rightarrow 48.30\%$), DTD ($45.90\% \rightarrow 48.01\%$), and Flowers ($68.26 \rightarrow 71.82\%\%$). These are precisely the regimes where prototype interference is most problematic: small angular overlaps between semantically related classes translate into large error rates. By softly encouraging orthogonality in the text encoder (rather than imposing it in one closed-form step), **ORION** spreads these classes apart while still preserving their relative structure to the image features, leading to consistent improvements across all 11 datasets.

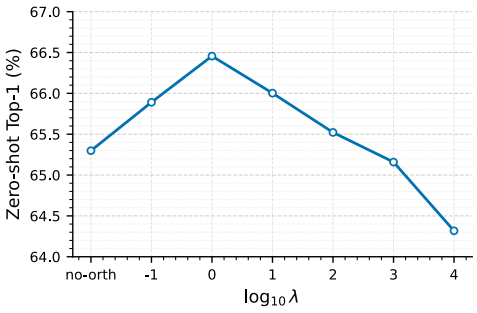

Figure 3. Effect of the orthogonality penalty weight λ on average zero-shot performance across 11 datasets. We sweep $\log_{10}\lambda \in \{-1, 0, 1, 2, 3, 4\}$, with "no-orth" corresponding to $\lambda = 0$. Moderate regularization ($\lambda \approx 2$) yields the best performance, while overly large values degrade accuracy due to excessive hardening of class directions.

C. Ablation on orthogonality penalty weight λ

In Figure 3, we examine the effect of the orthogonality penalty weight λ from Eq. (2) of the main paper, on zero-shot accuracy averaged across all 11 datasets. The performance improves steadily as soft orthogonality is introduced. Accuracy rises from 65.4% without orthogonality to 66.5% at $\log_{10}\lambda = 0$ (i.e., $\lambda = 2$), marking the highest point in the curve. A slightly stronger penalty ($\log_{10}\lambda = 1$, $\lambda = 20$) maintains comparable performance (66.1%). Beyond this range, $\lambda \in [2, 20]$, accuracy declines smoothly, e.g., 64.7% at $\log_{10}\lambda = 4$ ($\lambda = 20000$), as excessive orthogonality over-constrains the geometry, forcing embeddings toward near-uniform orthogonality. This unimodal and smooth behavior highlights the robustness of ORION. Moderate soft orthogonality reliably enhances angular dispersion without collapsing meaningful class relationship, in contrast to rigid whitening (SVD).

Table 9. Full per-dataset comparison of training-free SVD orthogonalization vs. ORION under MTA. Closed-form SVD whitening harms performance across most datasets, whereas our learnable orthogonality improves accuracy consistently.

Method	Pets	SUN397	Aircraft	DTD	EuroSAT	Cars	Food101	Flowers	Caltech101	UCF101	ImageNet	Avg
MTA (baseline)	88.24	66.67	25.32	45.90	45.36	68.47	84.95	68.26	94.21	68.11	69.11	65.87
+ SVD (closed-form)	80.40	65.40	20.40	46.20	40.90	60.20	82.80	57.10	92.10	63.80	64.20	61.23
+ ORION (ours)	**89.10**	**67.80**	**27.00**	**48.01**	**48.30**	**68.93**	**85.70**	**71.82**	**94.90**	**71.30**	**70.01**	**67.53**

Table 10. Integrating ORION in prompt augmentation with CuPL (ViT-L/14). Replacing the baseline text encoder in CuPL with **ORION** consistently improves zero-shot Top-1 accuracy across datasets.

Method	Pets	SUN397	Aircraft	DTD	EuroSAT	Cars	Food101	Flowers	Caltech101	UCF101	ImageNet	Avg
CuPL	93.8	73.31	36.11	61.7	56.6	77.6	93.36	79.7	93.5	78.36	76.7	74.60
CuPL w/ ORION	**94.2**$_{+0.39}$	**73.69**$_{+0.38}$	**36.6**$_{+0.49}$	**61.85**$_{+0.15}$	**58.67**$_{+2.11}$	**78.1**$_{+0.47}$	**93.7**$_{+0.34}$	**79.9**$_{+0.23}$	**95.9**$_{+2.45}$	**79.41**$_{+1.05}$	**77.21**$_{+0.52}$	**75.38**$_{+0.78}$

Table 11. ORION complements zero-shot text refinement (ZLaP) (ViT-B/16). Integrating **ORION** into ZLaP further improves performance, indicating that ORION enhances existing text-refinement pipelines.

Method	Pets	SUN397	Aircraft	DTD	EuroSAT	Cars	Food101	Flowers	Caltech101	UCF101	ImageNet	Avg
ZLaP	87.9	67.77	26.28	45.98	57.67	66.8	87.16	67.88	91.85	73.8	69.69	67.53
ZLaP w/ ORION	88.4$_{+0.50}$	67.75$_{-0.02}$	25.2$_{-1.08}$	50.35$_{+4.37}$	59.64$_{+1.97}$	68.93$_{+2.13}$	87.1$_{-0.06}$	75.11$_{+7.23}$	93.91$_{+2.06}$	71.61$_{-2.19}$	70.01$_{+0.32}$	68.91$_{+1.38}$

D. Comparison with prompt augmentation methods

We now assess the complementary nature of **ORION** on prompt augmentation methods such as CuPL [26]. CuPL expands the textual search space by leveraging external LLMs to generate diverse class descriptions, thereby improving coverage of semantic variations. Since **ORION** operates directly on the text encoder and refines the geometry of text prototypes, we initialize CuPL with **ORION**'s refined text-encoder and prototypes. Table 10 demonstrates that **ORION** can be seamlessly integrated into prompt-augmentation pipeline to further boost performance, with notable accuracy gains on EuroSAT (56.6% $\rightarrow$ 58.67%, +2.11) and UCF101 (78.36 $\rightarrow$ 79.41%, +1.05%).

E. Comparison to text refinement methods

We further evaluate the compatibility of **ORION** with zero-shot text refinement methods such as ZLaP[15], which adapt textual representations using unlabeled test data. While ZLaP already improves alignment through transductive updates, ORION operates by refining the geometry of class prototypes in the text embedding space. As shown in Table 11, combining the two yields consistent overall gains (+1.38% on average), with notable improvements on Flowers (67.88 $\rightarrow$ 75.11, +7.23), DTD (45.98 $\rightarrow$ 50.35, +4.37), and Cars (66.8 $\rightarrow$ 68.93, +2.13). These gains suggest that ORION enhances class separability in regimes where fine-grained or texture-based distinctions are critical. Overall, the results confirm that ORION remains broadly complementary to existing zero-shot text refinement strategies.