LiteEmbed: Adapting CLIP to Rare Classes

Supplementary Material

A. Appendix

In Section A.1, we present extended 1/2/4/8/16-shot results, highlighting how LiteEmbed scales with data availability and where gaps between our method and existing baselines are most pronounced. In Section A.2, we provide details of our data collection pipeline and the LLM prompts used to construct the NOVA benchmark. In Section A.3, we show additional results across CLIP backbones to demonstrate the architecture-agnostic nature of our approach. In Section A.4, we compare against Textual Inversion to illustrate the limitations of generation-oriented personalization when applied to large-scale discriminative settings. In Section A.5, we further analyze a reference-image baseline to clarify why operating purely in image space undermines downstream compositionality. Finally, we include expanded qualitative examples across retrieval, segmentation, detection, and generation tasks in Section A.6 as well as additional analysis of our PCA-based subspace choice in Section A.7.

A.1. 1/2/8/16 few shot results

To further study how LiteEmbed scales with data availability, we compare its performance against CoOp [6], MaPLe [4], and TIP-Adapter-FTIP-Adapter-F [5] under 1-, 2-, 4-, 8-, and 16-shot settings (shown in Figure 1). LiteEmbed consistently outperforms these baselines even in the 1- and 2-shot regimes, where the gap is especially pronounced. On datasets such as Indian Food, the improvement over prior methods reaches 15–20% in the extreme low-shot setting, and on Indian Actors, the margin widens further to 25–35%, highlighting LiteEmbed's strong generalization under minimal supervision. While TIP-Adapter-F begins to close the gap as more samples become available, it still trails LiteEmbed across all shot counts. Moreover, existing approaches remain tied to fixed few-shot setups and do not naturally extend to incremental integration of new concepts.

A.2. Dataset scraping details

The five datasets in our NOVA benchmark, namely *Indian Singers*, *Indian Actors*, *Game Characters*, *Fashion Outfits*, and *Landmarks*, were newly collected to evaluate CLIP's performance on post-2023 emerging content and culturally-specific domains. We developed a systematic data collection pipeline using iCrawler [1] with Bing as the image source, chosen for its reliability and comprehensive content coverage. To ensure our benchmark targets content outside CLIP's pretraining distribution (which has a knowledge cutoff around 2021), we generated class lists using the following prompt with a large language model [2]:

> Generate a list of 50 domain-specific classes that emerged or gained significant prominence after 2023, for each of Indian Singers, Indian Actors, Game Characters, Fashion Outfits, and Landmarks. Focus on entities that:
> 1. have substantial visual presence online,
> 2. represent diverse subcategories within the domain,
> 3. are culturally specific or region-specific where applicable,
> 4. are unlikely to have been well-represented in pre-2022 vision-language training datasets.
>
> For Indian Singers and Actors, prioritize emerging talents from recent films and music releases. For Game Characters, focus on notable characters popular games released in 2023-2024. For Fashion Outfits, select outfits from emerging fashion brands and designers that launched or gained international recognition post-2023. For Landmarks, include architectural structures and public installations completed after 2023.

A.3. Ablation across CLIP backbones

To demonstrate the generalizability of our approach across different model scales, we evaluate on both ViT-B/32 and ViT-L/14 CLIP backbones (Table 1). Despite their substantial differences in model size and patch resolution (32×32 vs. 14×14), LiteEmbed achieves consistent improvements across both architectures: +33.3% for ViT-B/32 and +35.75% for ViT-L/14, averaged over all the datasets. This backbone-agnostic behavior demonstrates that our method inherits the architecture's flexibility without requiring model-specific tuning.

A.4. Textual Inversion for classification

While personalization methods like Textual Inversion [3] have demonstrated remarkable success in embedding new visual concepts directly into CLIP's vocabulary for generation tasks, their efficacy in downstream discriminative tasks remains underexplored. To investigate this, we conduct an incremental classification experiment on the Indian Food dataset (80 classes), comparing the zero-shot classification performance of base CLIP ViT-L/14 text embeddings against learned TI embeddings. Our experimental protocol

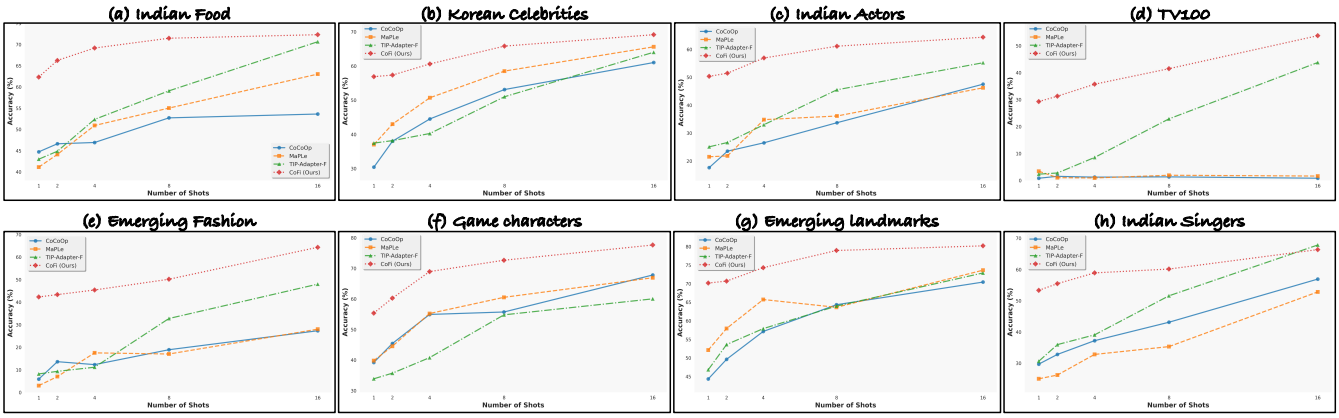

Figure 1. Comparison of few-shot classification accuracy across 1-, 2-, 4-, 8-, and 16-shot settings on multiple datasets.

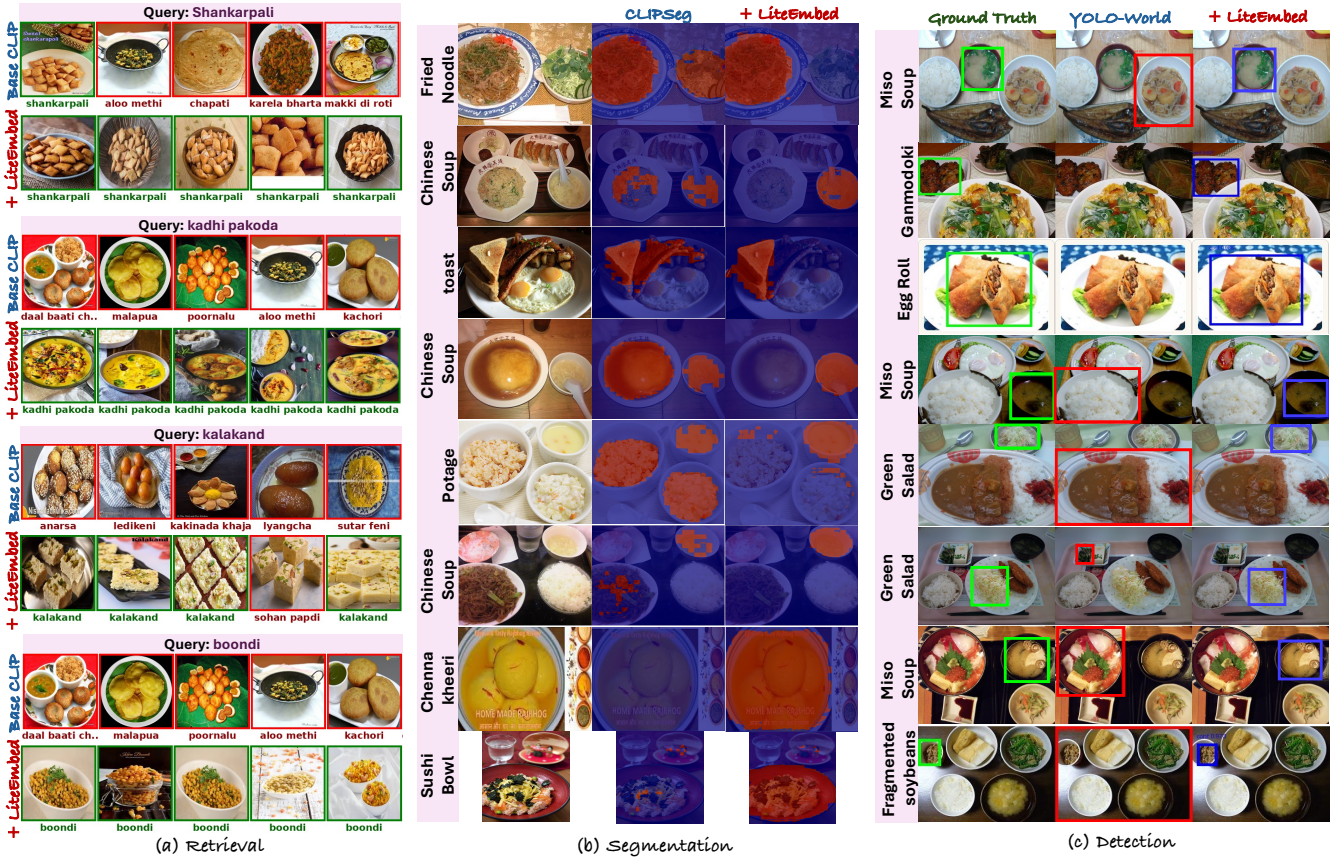

Figure 2. Additional Qualitative results across downstream tasks. LiteEmbed consistently improves retrieval, segmentation, and detection quality over baseline CLIP embeddings.

follows an incremental learning setup: for each task k $\in$ [1, 80], we evaluate classification accuracy on k classes using both base CLIP embeddings ("a photo of [class]") and personalized TI embeddings ("a photo of <class>") as text encoders. As shown in Figure 3, we observe that TI embeddings provide substantial gains (15-22% improvement) for small vocabularies (8-10 classes), but this advantage rapidly diminishes as the number of classes grows. Beyond approximately 13 classes, base CLIP consistently outperforms TI, with the performance gap widening to 9.3% at the full 80-

Table 1. Classification Performance (Top-1 Accuracy %): Baseline CLIP vs Our Textual Inversion Method

Dataset	ViT-B/32		ViT-L/14	
	Baseline CLIP	LiteEmbed	Baseline CLIP	LiteEmbed
Indian Food	36.05	63.22	44.21	71.28
Game Characters	25.31	64.13	31.15	69.21
Indian Actors	18.91	51.12	21.80	59.34
Indian Singers	23.46	55.67	31.85	60.23
Landmarks	37.38	74.12	47.81	89.18
TV100	2.38	34.11	3.51	38.13
Fashion Outfits	7.68	42.38	9.31	51.16
Korean Celebrities	25.23	54.61	27.16	64.43
Average	**21.59**	**54.92**	**27.12**	**62.87**

class setting (44.2% vs 34.9% accuracy).

This trend suggests that while TI's generation-optimized embeddings capture fine-grained visual details essential for reconstruction, they lack the semantic structure and inter-class relationships inherent in CLIP's pre-trained representations, which prove critical for robust discrimination across large vocabularies. These findings highlight a fundamental trade-off: embeddings optimized for generation (via diffusion reconstruction loss) may not transfer effectively to discriminative tasks, suggesting the need for multi-objective training strategies that balance both generative fidelity and discriminative power.

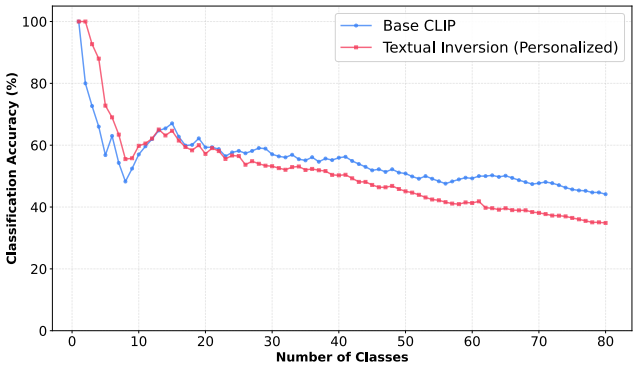

Figure 3. Textual Inversion for classification.

A.5. Using reference images directly

We evaluate a few-shot reference-based baseline that bypasses CLIP's text encoder entirely by directly using mean image embeddings from four reference images as class representatives. As shown in Table 2, this few-shot approach yields only a modest average gain of +4.04% across seven diverse datasets and behaves inconsistently, even reducing performance relative to base CLIP on three of the seven datasets (Indian Food, Indian Singers, and Landmarks). Most of the apparent improvements occur on datasets where CLIP's original text embeddings perform extremely poorly (e.g., 3.49% on TV100, 7.68% on emerging fashion), suggesting that while this method can offer small boosts when text-based representations fail, it cannot meaningfully improve performance beyond that narrow regime.

In contrast, retrieval benefits more reliably, since image-to-image similarity is often stronger than text-to-image alignment for visual search. However, this baseline fundamentally breaks CLIP's vision–language alignment by operating entirely in image space, making it incompatible with downstream tasks that rely on text–image compositionality, such as open-vocabulary detection, referring segmentation, and text-to-image generation. It also cannot take advantage of linguistic compositionality (e.g., combining "red" and "car") or zero-shot generalization to unseen text prompts—both central to CLIP's utility.

By comparison, LiteEmbed learns embeddings directly within CLIP's native text space, preserving full compositionality and compatibility with the text encoder while providing consistent improvements across tasks.

A.6. Additional qualitative results for all tasks

We show additional results in Figure 4 and 2 demonstrating LiteEmbed's strong adaptability to downstream tasks like retrieval, segmentation, detection and generation.

A.7. Choice of k (split) in PCA

We set $k \geq 2$ (i.e., we drop the first principal component) for the fine subspace. To justify this choice, we ran a PCA analysis on 75 classes drawn from five broad categories–dogs, cats, vehicles, food, and buildings. For each principal component, we quantified how much it favors coarse-grained (across categories) versus fine-grained (within a category) separation by taking the ratio of average cross-category distances to within-category distances.

PC1 shows a ratio of 5.02, meaning it separates classes across different categories about 5× more strongly than it separates classes within the same category. By contrast, PCs 2–5 have an average ratio of 1.62, suggesting they are far less biased toward coarse distinctions.

Table 2. Classification and Retrieval Performance using ViT-B/16 with 4-shot Reference Image Embeddings

Dataset	**Classification (Top-1 Acc %)**		**Retrieval (P@5 %)**	
	Baseline	Few-shot	Baseline	Few-shot
Indian Food	39.10	26.86	52.00	58.75
Game Characters	29.86	42.76	49.85	59.38
Indian Actors	23.06	34.48	27.20	51.60
Indian Singers	31.13	28.79	39.11	42.00
Landmarks	45.31	42.10	60.57	69.71
TV100	3.49	17.51	6.60	35.80
Fashion Outfits	7.68	16.29	30.91	45.82

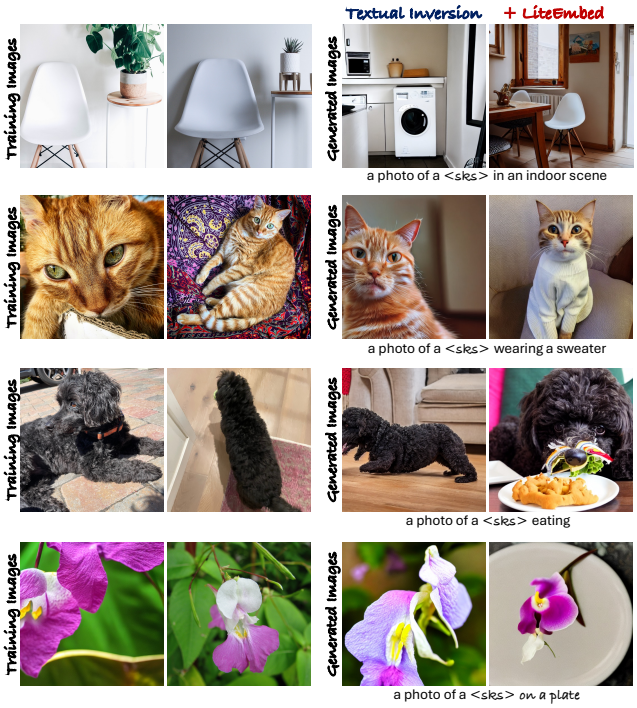

Figure 4. Additional Qualitative results across downstream tasks. LiteEmbed consistently improves retrieval, segmentation, and detection quality over baseline CLIP embeddings.

We also inspected the ten class pairs that PC1 separates the most, and all of them turned out to be cross-category (for example, *cat vs. building*). When we evaluated the same pairs on PCs 2–5, their separation dropped by 76%, meaning that without PC1, those pairs lose most of their discriminability. The reverse pattern also holds: within-category pairs that are highly separated on PC2 retain only 22% of that separation when projected onto PC1.

Taken together, these results show a clear asymmetry: PC1 mainly captures broad, category-level differences, whereas PC2 and later components encode the fine-grained variations that matter for personalized classification. This is why we exclude PC1 and rely on PCs $k \geq 2$ for our task.

References

[1] icrawler. https://github.com/hellock/icrawler. 2

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[3] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *International Conference on Learning Representations (ICLR)*, 2023. 2

[4] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19113–19122, 2023. 2

[5] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European Conference on Computer Vision (ECCV)*, pages 493–510. Springer, 2022. 2

[6] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16816–16825, 2022. 2