CoPA: Compressing Visual In-Context Learning for Parameterized Test-Time Adaptation

Anonymous authors
Paper under double-blind review

Abstract

We study visual few-shot adaptation from labeled support examples. CoPA processes a support set with a frozen vision-language model, compresses the resulting activations with a Perceiver resampler, and uses an episodically meta-trained hypernetwork to generate low-rank updates for query inference. The backbone remains frozen and no gradients are taken at inference time. Across 14 datasets and four backbones, we evaluate predictive performance for up to 16 support examples. In this range, CoPA outperforms the frozen prompting and one-step test-time optimization baselines evaluated here. These comparisons do not isolate the effect of the architecture because CoPA also receives supervised episodic training. A separate profile up to 64 supports shows that query-phase memory and latency are approximately independent of support-set size, while support processing still grows with that size. We characterize CoPA as an amortized, support-conditioned few-shot adapter, rather than a training-free in-context learning method or an unlabeled test-time adaptation method.

1 Introduction

Recent work has shown that language-model in-context learning (ICL) can continue to improve as the number of demonstrations grows into the hundreds or thousands Brown et al. (2020); Li et al. (2023); Agarwal et al. (2024). This scaling does not transfer cleanly to vision-language models (VLMs). As illustrated in Figure 1, frozen visual ICL is frequently *non-monotonic*: accuracy may rise at a few shots and then degrade or saturate once many images occupy the context, a pathology we refer to as visual context overload.

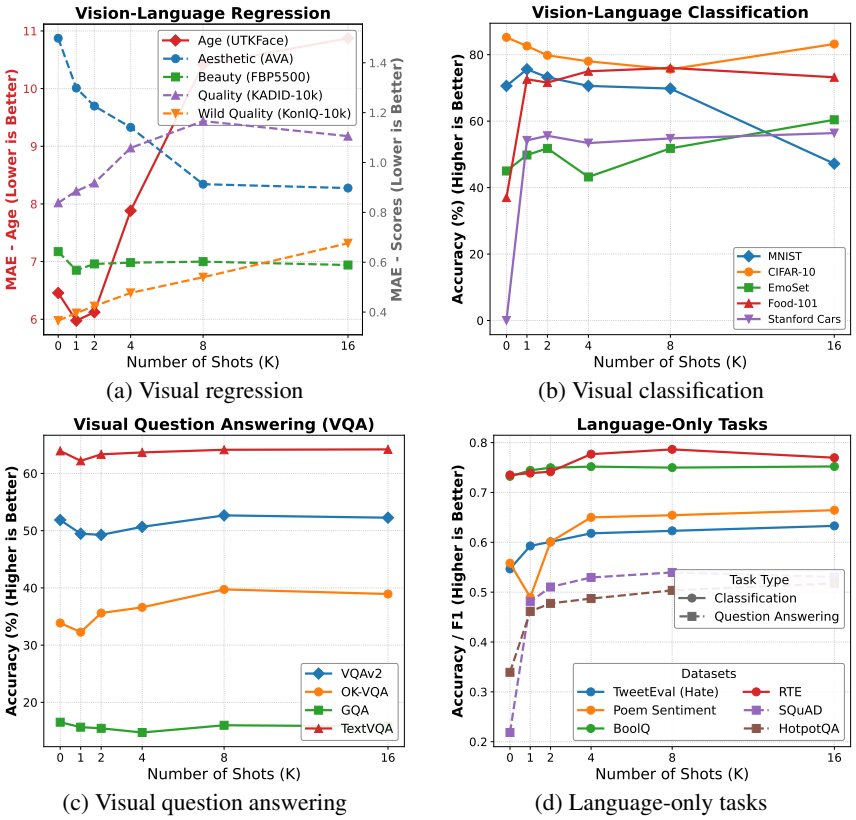
A natural alternative is to fold the support set into parameters rather than into the prompt. Gradient-based test-time training (TTT) does this by optimizing fast weights at inference Akyürek et al.; Hu et al. (2025), but backpropagation over image-token contexts is slow and memory-heavy. We instead *amortize* that map. CoPA (Compressive Parameter Adaptation) runs a frozen support forward pass, compresses the cached activations with a Perceiver-style resampler, and uses a HyperNetwork to emit low-rank (LoRA) updates Hu et al. (2021) for a sparse subset of the frozen backbone. The support tokens are then discarded, so query generation no longer attends over K images.

This protocol is more specific than generic "test-time adaptation." CoPA assumes (i) a labeled support set, (ii) a HyperNetwork ϕ that is episodically meta-trained, and (iii) query-independent fast weights. It is closest to support-conditioned few-shot adaptation / visual ICL compression Min et al. (2022); Ha et al. (2016), not to unlabeled TTA. We evaluate it as such: frozen ICL and one-step TTT remain important reference points, but they do not receive the same supervised meta-training budget as ϕ.

Contributions.

- **Amortized support-to-weight adaptation.** We specify a demonstration-to-LoRA architecture that compresses variable-length visual supports into a fixed latent budget and injects episode-specific fast weights into a frozen VLM, with no inference-time gradients.
- **Stable few-shot visual ICL for $K \leq 16$.** On 14 datasets spanning regression, classification, and VQA, CoPA avoids the non-monotonic frozen-ICL trend in the evaluated range and remains high from $K = 1$ onward, rather than requiring long concatenated contexts.

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.



(a) Visual regression (b) Visual classification

(c) Visual question answering (d) Language-only tasks

Figure 1: **Frozen ICL versus number of shots (K).** The visual tasks show task-dependent degradation or saturation over $K \leq 16$; language-only tasks provide a reference. Larger K is profiled only for cost.

- **Query-phase cost independent of K.** After the one-time support encoding, query memory and prefill latency are constant in K (profiled to $K = 64$), whereas frozen ICL grows with context length and gradient TTT OOMs.

2 Related Work

Scaling and mechanisms of in-context learning. Language models can learn from demonstrations in context Brown et al. (2020); Lu et al. (2022); Todd et al. (2023), and longer-context studies extend this setting to many examples Li et al. (2023); Agarwal et al. (2024). Multimodal scaling can be less reliable Jiang et al. (2024); approaches to context dilution include filtering demonstrations or tokens Yuan et al. (2024) and changing the training distribution Dong et al. (2025). Episodic training can improve in-context adaptation Min et al. (2022), while multimodal task vectors suggest that support examples can also be summarized in activations Huang et al. (2024). Link-context evaluations further distinguish using a support-to-query relationship from merely recognizing familiar examples Tai et al. (2024). CoPA studies a different representation of labeled visual supports: episode-specific model weights.

Test-time optimization. Adapting parameters for an incoming example has roots in transductive and local learning Joachims et al. (1999); Bottou & Vapnik (1992); Lee et al. (2020); Hu et al. (2020); Liu et al. (2018). Recent test-time training methods optimize temporary weights from available context Zhang et al. (2025); Tandon et al. (2025); Akyürek et al.; 2025); Hu et al. (2025). Their objectives and applications include task-specific adaptation Chollet (2019), entropy-based adaptation Wang et al. (2020); Niu et al. (2022); Chen et al. (2024b); Niu et al. (2023); Liang et al. (2025), and vision-language adaptation Kojima et al. (2025); Fan et al. (2025). Our one-step TTT imple-

mentations are reference points, not supervision-matched or fully tuned competitors: CoPA instead learns its support-to-weight mapping through labeled episodic meta-training and does not optimize weights at inference time.

Hypernetworks and low-rank adaptation. CoPA uses the low-rank update parameterization of LoRA Hu et al. (2021). Hypernetworks generate model parameters from conditioning information Ha et al. (2016); related methods generate adapters from text or contextual representations Liu et al. (2026); Charakorn et al. (2025); Shao et al. (2025). Visual and embodied applications also use hypernetworks to generate task-dependent parameters Akinwande et al. (2024); Xiong et al. (2025). CoPA combines support activation capture, fixed-budget resampling, and a hypernetwork that emits low-rank updates for a frozen VLM. We claim this combination and its evaluated behavior, rather than a new parameter-generation primitive.

Context compression and inference-time compute. Visual context can be shortened or transformed before inference Zhou et al. (2024); Fu et al. (2025); Cheng et al. (2025); Chen et al. (2024a). Other methods spend additional inference-time compute on reasoning or selection Wu et al. (2024); McCoy et al. (2024); Wei et al. (2022); Wang et al. (2022); Snell et al. (2025); Damani et al. (2024); Yao et al. (2023). CoPA instead processes labeled supports once to generate temporary weights and removes those supports from the query prompt. Its query-phase cost is independent of support count, but its support-processing cost is not.

3 METHOD

3.1 PROBLEM SETUP: SUPPORT-CONDITIONED VISUAL ICL

We study K-shot visual in-context learning (V-ICL) for multimodal autoregressive transformers. Following the episodic protocol of MetaICL Min et al. (2022), each episode consists of a *labeled* support set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{K}$ and a query pair (x_q, y_q). A frozen VLM f_θ can either (a) condition on an interleaved support prompt (standard ICL) or (b) receive a parameter update computed from $\mathcal{S}$.

CoPA takes option (b). We learn a HyperNetwork g_ϕ that maps the support episode to low-rank weight updates,

$$\Delta\Theta \;=\; g_\phi(\mathcal{S}; \theta), \tag{1}$$

while θ remains frozen at both meta-training and inference (Figure 2). Only ϕ (Perceiver and HyperNetwork) is trained, using teacher-forced query answers on labeled episodes. At test time the same feed-forward map is applied once to $\mathcal{S}$; no gradients are computed.

What this is, and is not. CoPA is parameterized in the sense that the query is answered by an adapted backbone, and it is test-time in the sense that $\Delta\Theta$ is episode-specific and discarded after the query. It is not unlabeled TTA, not training-free ICL, and not a claim that a single ϕ is universal across all visual tasks. All comparisons in Section 4 should be read with this protocol difference in mind: CoPA receives supervised episodic meta-training; frozen ICL and the TTT baselines do not.

3.2 FAST-WEIGHT PARAMETERIZATION (LORA GENERATION)

We adapt a sparse set of linear projections inside the transformer blocks (FFN projections) designated by an index set $\mathcal{L}$. For a targeted frozen linear map with weight $W^{(\ell)} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and bias $b^{(\ell)}$, we inject a rank-r additive update using the LoRA factorization Hu et al. (2021) $\Delta W^{(\ell)} = B^{(\ell)} A^{(\ell)}$. The adapted layer is

$$\hat{\text{Linear}}^{(\ell)}(h) = h(W^{(\ell)})^\top + b^{(\ell)} + s^{(\ell)} \cdot h(\Delta W^{(\ell)})^\top, \tag{2}$$

where $s^{(\ell)} \in \mathbb{R}$ is a learned scalar gate predicted by the HyperNetwork. The HyperNetwork generates episode-specific weights $(A^{(\ell)}, B^{(\ell)}, s^{(\ell)})$ for each batch item. Unless stated otherwise we adapt a single site, the FFN down-projection at layer 8; Appendix H compares alternative sites, including the stronger but less conservative up-projection.

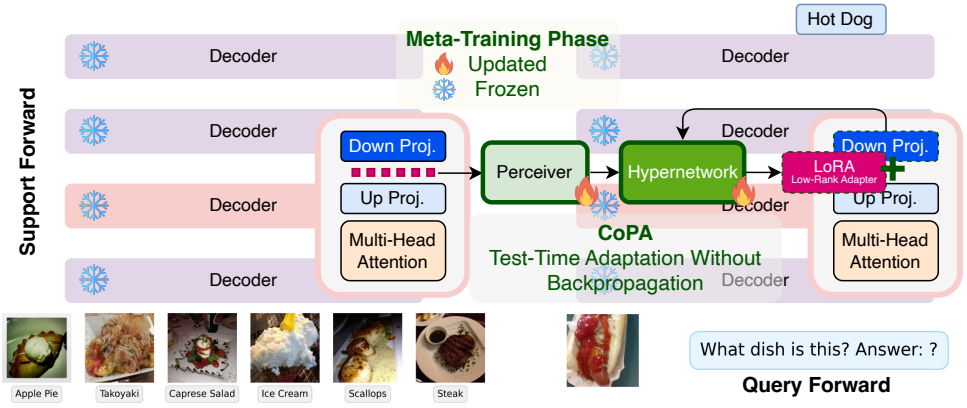


Figure 2: **Overview of CoPA.** Only the Perceiver and HyperNetwork are meta-trained (green); the VLM backbone stays frozen. For each episode, a no-grad support pass caches input activations at the adapted layer. The Perceiver compresses this variable-length context into M latent tokens; the HyperNetwork maps them to LoRA fast weights that are injected for the query pass. The support tokens are then discarded, so query compute does not grow with K.

3.3 Support-to-Fast-Weight Compression

Rather than encoding raw support images, we use the frozen VLM as a feature extractor. During a **support forward pass** (no gradients), each injected layer ℓ caches its input activations $H_{\text{in}}^{(\ell)} \in \mathbb{R}^{B\times T\times d_{\text{in}}}$, representing pre-activation features for the sequence of length T. We define our layer-specific context directly from these features: $C^{(\ell)} = H_{\text{in}}^{(\ell)}$.

The hypernetwork g_ϕ then maps this context and the frozen target weights to the fast weights via three main components (full architectural and initialization details are provided in Appendix B):

1. Perceiver Resampler: Token length T varies with K. To obtain a fixed-size representation, we project $C^{(\ell)}$ to a fixed width d and process it through a Perceiver-style resampler. This compresses the variable-length context into a fixed number of M learned latent task tokens $Z^{(\ell)} \in \mathbb{R}^{B\times M\times d}$, bounding the memory footprint.

2. Layer Identity Encoder: A single shared hypernetwork must generate distinct updates for different layers. We compute a layer-identity embedding $e^{(\ell)} \in \mathbb{R}^{1\times d}$ by linearly compressing and pooling the frozen weight matrix $W^{(\ell)}$.

3. Layer-Aware Residual Decoder: Finally, a residual decoder initializes a query vector that cross-attends first to the layer identity $e^{(\ell)}$, and then to the task tokens $Z^{(\ell)}$. This conditions the query on both *where* it is being injected and *what* task it must solve, projecting the refined query into the final parameters $A^{(\ell)}$, $B^{(\ell)}$, and $s^{(\ell)}$.

3.4 Training: Episodic Supervised Meta-Optimization

We optimize ϕ using labeled episodes, supervising the model via teacher-forced next-token prediction on the rendered ground-truth answer. Rather than updating the backbone θ, our episodic optimization updates only the hypernetwork ϕ and the Perceiver resampler (Figure 2). The process involves three steps: **(1) Support Forward (Frozen):** Run the frozen VLM on the support prompt under `no_grad` and cache inputs $H_{\text{in}}^{(\ell)}$. **(2) Generate Fast Weights:** For each adapted layer, form $C^{(\ell)}$, compute the fast weights via g_ϕ, and dynamically inject them. **(3) Query Forward & Loss:** Run the query prompt, compute cross-entropy exclusively on the completion tokens (masking prompt tokens), and backpropagate through the injected fast-weight path to update ϕ. At test time, the fast weights are episode-specific and discarded after answering, enabling parameter-

ized adaptation without permanently modifying the backbone. (See Appendix F and C for exact prompt formatting and efficient batch training implementations).

3.5 Complexity Analysis

Let T_s denote the support-prompt token count and T_q the query-prompt token count. CoPA first runs the frozen VLM over the support prompt and then applies Perceiver cross-attention with cost proportional to MT_s per resampling layer, apart from width-dependent factors. The Perceiver does not eliminate the backbone's support-pass compute or its peak support-pass activation memory.

After support processing, CoPA discards the support KV cache and retains fixed-size episode-specific adapter weights. Consequently, for fixed T_q, backbone, adapter sites, and rank, query-phase memory and compute do not grow with T_s. This is a query-phase property, not a constant-cost end-to-end inference claim. Table 2 reports the support and query phases separately. Their measured times can be added for the profiled single-query setting, but we have not benchmarked amortization across multiple queries that share one support set.

4 Experiments

We evaluate CoPA as a support-conditioned few-shot adapter on 14 datasets and four vision-language models, and compare it to frozen ICL and one-step gradient TTT.

4.1 Experimental Setup

Datasets & Benchmarks. We curate a comprehensive benchmark suite comprising 14 datasets across three modalities. **Fine-Grained Regression:** We evaluate the mapping of visual features to continuous scalars—a historically challenging task for standard VLMs—on Age Prediction (UTK-Face Zhang et al. (2017)), Aesthetic Assessment (AVA Murray et al. (2012), SCUT-FBP5500 Liang et al. (2018)), and Image Quality Assessment (KonIQ-10k Hosu et al. (2020), KADID-10k Lin et al. (2019)). **Few-Shot Classification:** We test semantic categorization across diverse domains, including affective computing (EmoSet Yang et al. (2023)), fundamental object recognition (CIFAR-10 Krizhevsky et al. (2009), MNIST Deng (2012)), and fine-grained visual categorization (Stanford Cars Krause et al. (2013), Food-101 Bossard et al. (2014)). **Visual Question Answering (VQA):** We assess open-ended reasoning using VQAv2 Antol et al. (2015), OK-VQA Marino et al. (2019), GQA Hudson & Manning (2019), and TextVQA Singh et al. (2019).

Standardized Prompting. To ensure a fair comparison and disentangle model capability from instruction-following variance, we enforce strict output formatting. For regression, we prompt models to output only numerical values (e.g., "[Image] Rate the image quality (0-5). Only output a number."). For classification, models predict the exact class index or category name. For VQA, we append "Answer the following question using a single word or a short phrase" to constrain verbosity.

VLM Backbones. We validate model agnosticism by implementing CoPA across four diverse architectures: *Gemma 3 4B-it* Team et al. (2025), *Qwen 2.5 7B* Bai et al. (2025), *Phi-3.5-vision (4.2B)* Abdin et al. (2024), and *Idefics3-8B-Llama3* Laurençon et al. (2024). All experiments are performed using Gemma 3 4B-it unless otherwise specified.

Baselines and comparison scope. We compare CoPA against two inference-time paradigms that do *not* meta-train an auxiliary network on labeled episodes. **In-context learning (ICL)** leaves the backbone frozen and concatenates demonstrations in the prompt. We evaluate *Random* sampling from the training pool and *Retrieval-Augmented kNN* using SigLIP embeddings. For classification we use class-balanced kNN to avoid majority-class collapse; for regression and VQA, kNN retrieves the K nearest visual neighbors. **Test-time training (TTT)** optimizes LoRA adapters at inference. *TTT (Support Answers)* Akyürek et al. is a supervised one-step update on the ground-truth answer tokens of the support set. *TTT (Query Context)* Hu et al. (2025) is an unsupervised one-step update that minimizes causal LM loss on the query prompt text. Image tokens are masked from both losses, following the multimodal constraint that next-token CE is undefined on visual embeddings. We use the same LoRA rank ($r = 8$) and adaptation site for CoPA and TTT, but the methods do not

have matched training budgets or matched adaptation compute. TTT takes one inference-time SGD step, whereas CoPA uses a hypernetwork trained beforehand on labeled episodes. Our TTT results therefore characterize this specific lightweight implementation, not the best performance achievable with tuned or multi-step multimodal test-time optimization.

CoPA is not a third training-free baseline. Its HyperNetwork is episodically trained on labeled support/query pairs from the task's training split (or from a specified mixture in the transfer study). Gains relative to frozen ICL therefore confound the adaptation architecture with additional supervised training. Section 4.5 examines how CoPA responds to different sources of meta-training data; it does not isolate the architectural contribution from the training signal.

Table 1: **Main results averaged over $K \in \{1, 2, 4, 8, 16\}$.** Regression: MAE ↓. Classification and VQA: accuracy (%) ↑. TTT-C and TTT-S denote query-context and support-answer one-step TTT. CoPA-M is one model trained jointly on Age, MNIST, and VQAv2. For the CoPA-M row, blue denotes a target dataset included in meta-training and red denotes a target dataset excluded from meta-training. CoPA is episodically meta-trained; ICL and one-step TTT are not.

Regression: MAE ↓						
Method	Age	Aesth.	Beauty	Q. Mod.	Q. Wild	Avg.
Zero-shot	7.83	1.95	0.48	0.87	0.48	2.32
Random selection						
ICL	9.24	1.73	0.76	1.06	0.60	2.68
TTT-C Hu et al. (2025)	9.24	1.72	0.76	1.06	0.60	2.68
TTT-S Akyürek et al.	9.25	1.73	0.76	1.06	0.60	2.68
CoPA	**5.24**	**0.50**	**0.30**	**0.46**	**0.25**	**1.35**
CoPA-M	6.12	1.68	1.04	1.77	1.35	2.39
kNN selection						
ICL	7.27	1.60	0.40	1.07	0.61	2.19
TTT-C Hu et al. (2025)	7.26	1.61	0.40	1.07	0.61	2.19
TTT-S Akyürek et al.	7.27	1.60	0.40	1.07	0.60	2.19
CoPA	**5.22**	**0.50**	**0.30**	**0.46**	**0.25**	**1.35**
Classification: accuracy (%) ↑						
Method	MNIST	EmoSet	Cars	Food	CIFAR-10	Avg.
Zero-shot	67.20	50.40	0.00	35.10	85.20	47.58
Random selection						
ICL	63.60	52.74	53.80	73.18	79.38	64.54
TTT-C Hu et al. (2025)	63.44	52.64	53.68	73.20	79.38	64.47
TTT-S Akyürek et al.	63.72	52.80	53.78	73.12	79.46	64.58
CoPA	**95.34**	**77.44**	**79.50**	**85.66**	**92.66**	**86.12**
CoPA-M	80.50	54.20	33.50	67.70	87.40	64.66
kNN selection						
ICL	56.72	53.08	74.32	79.46	78.66	68.45
TTT-C Hu et al. (2025)	56.44	53.18	74.32	79.44	78.74	68.42
TTT-S Akyürek et al.	56.72	53.08	74.50	79.42	78.78	68.50
CoPA	**95.34**	**77.42**	**79.54**	**85.74**	**92.68**	**86.14**
VQA: accuracy (%) ↑						
Method	VQAv2	OK-VQA		GQA	TextVQA	Avg.
Zero-shot	55.20	34.83		48.30	65.70	51.01
Random selection						
ICL	55.69	36.83		46.80	64.88	51.05
TTT-C Hu et al. (2025)	55.75	36.53		47.01	64.96	51.06
TTT-S Akyürek et al.	55.61	36.63		46.95	64.93	51.03
CoPA	**72.81**	**54.69**		**54.99**	**71.85**	**63.59**
CoPA-M	72.43	44.77		54.81	68.33	60.09
kNN selection						
ICL	53.39	36.67		46.47	65.44	50.49
TTT-C Hu et al. (2025)	53.63	36.21		46.29	65.53	50.41
TTT-S Akyürek et al.	53.71	36.59		46.35	65.41	50.52
CoPA	**72.77**	**54.63**		**54.81**	**71.73**	**63.48**

Implementation details. Unless otherwise stated, CoPA uses $K = 16$ shots, LoRA rank $r = 8$, and $M = 128$ Perceiver latents. We inject adapters into the FFN down-projection at layer index

$\mathcal{L} = \{8\}$. Appendix H shows that the FFN up-projection at the same layer is stronger on the two probed tasks (Age MAE 5.04 vs. 5.21; EmoSet 78.1% vs. 76.4%). We keep down-projection as the main-paper default because it was the stable, conservative site used across all 14 datasets and four backbones, not because it is optimal. The TTT baselines use the same rank and site.

4.2 Main Results

We evaluate performance averaged across $K \in \{1, 2, 4, 8, 16\}$ shots to capture a holistic view of each method's context utilization. To disentangle the benefits of better demonstration selection from the benefits of our architectural adaptation, we report results using both *Random* selection and *k-Nearest Neighbors (kNN)* retrieval based on visual similarity (SigLIP embeddings). Furthermore, we include an Out-of-Distribution (OOD) study to test the cross-task generalizability of a single, jointly-trained CoPA model.

Reading the aggregate results. Each task entry in Table 1 averages performance over the five reported shot counts, so it summarizes the evaluated range rather than performance specifically at $K = 16$. The "Avg." column then averages the displayed datasets within one task family. In particular, the regression average combines MAEs from targets with different scales; it is a compact descriptive summary, not a scale-normalized measure of improvement. We therefore use individual dataset columns when discussing transfer or retrieval effects. Small differences between frozen ICL and one-step TTT, often visible only at the second decimal place, should likewise not be interpreted as evidence of a reliable advantage without uncertainty estimates. The larger CoPA gaps remain comparisons across different training protocols.

Analysis. Table 1 should be read as a cross-paradigm comparison. Frozen ICL and one-step TTT are nearly identical across modalities, often at the second decimal place. The near-equality of frozen ICL and the one-step TTT baselines is an observation about these particular implementations. Although the TTT loss is computed on text tokens, this fact alone does not establish why its update has little effect on predictions. CoPA reads captured support activations and produces an adapter using a separately meta-trained hypernetwork. The comparison therefore cannot attribute the accuracy gap solely to the Perceiver, parameter generation, or the absence of inference-time gradients.

Retrieval is *not* a universal amplifier. kNN helps most regression and classification settings (random ICL average MAE $2.68 \rightarrow 2.19$; classification $64.54\% \rightarrow 68.45\%$), but it can hurt VQA: VQAv2 frozen ICL drops from 55.69 (random) to 53.39 (kNN), and GQA from 46.80 to 46.47. CoPA itself is almost insensitive to random vs. kNN in Table 1, which is consistent with compressing supports into a fixed-size task vector rather than attending over the raw prompt.

CoPA-M uses a single ϕ meta-trained on Age, MNIST, and VQAv2. "Out-of-distribution" in this row means that the evaluated dataset was excluded from meta-training; the model still receives labeled support examples from that dataset at inference time. Its results therefore measure adaptation to an unseen training dataset with labeled supports, not adaptation without target-task examples. Transfer varies by task: the VQA scores remain above random-selection ICL, while Cars and Food fall below it. Several regression scores are also worse than their corresponding frozen-ICL scores. Appendix G examines how the meta-training mixture affects this behavior; it does not isolate the architecture from the benefit of additional supervision.

4.3 Shot-count and cross-model results

Figure 3 compares frozen ICL and CoPA for $K \in \{1, 2, 4, 8, 16\}$. CoPA remains relatively stable over this evaluated range, but the figure does not establish accuracy scaling at hundreds of shots or show whether additional supports are used. The $K = 64$ measurement in Section 4.6 concerns cost only.

Backbones. Figure 4 reports VQAv2 results for Gemma 3 4B-it, Phi-3.5-vision, Qwen 2.5 7B, and Idefics3-8B. CoPA improves on the evaluated frozen ICL and one-step TTT implementations across these backbones. This tests several architectures, not universal architecture independence.

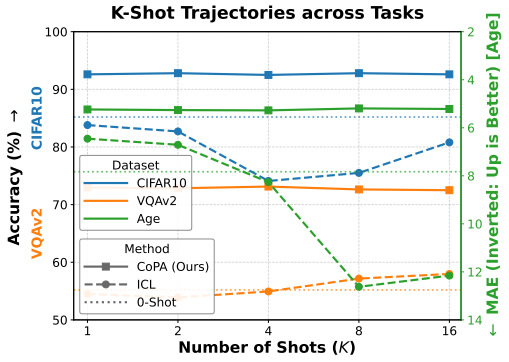


Figure 3: *Accuracy vs.* $K \leq 16$. Frozen ICL (dashed) is non-monotonic or saturating; dotted lines are zero-shot. CoPA (solid) is high at $K = 1$ and stable through $K = 16$. This supports shot-stable adaptation in the evaluated range, not a many-shot scaling law.

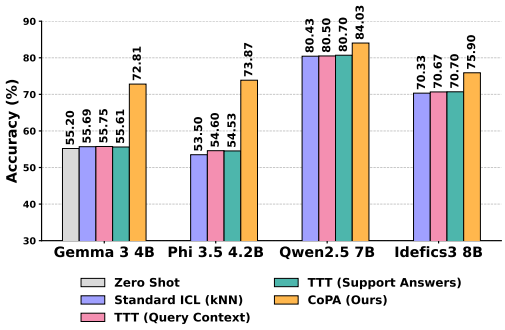


Figure 4: *Cross-model results on VQAv2.* CoPA improves all four backbones relative to frozen ICL and our one-step TTT baselines. Relative gains are larger on smaller models (Gemma 3, Phi-3.5-vision); stronger backbones (Qwen 2.5, Idefics3) reach the highest absolute accuracy.

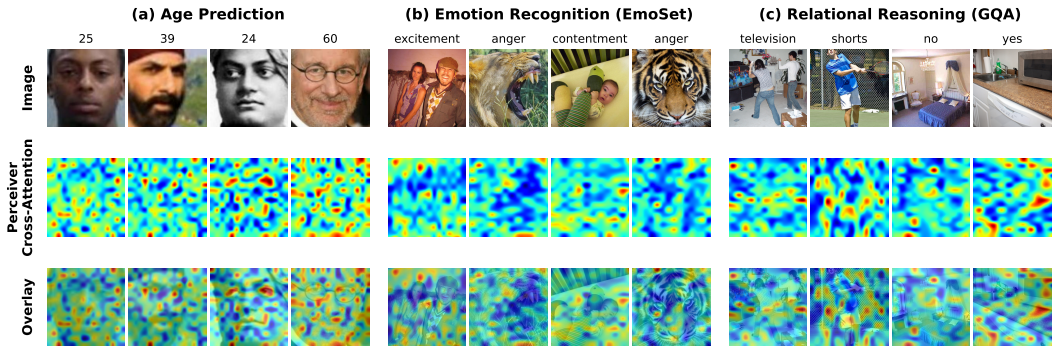


Figure 5: *Visualization of Perceiver cross-attention for support examples.* Four examples are shown for Age prediction (a), EmoSet (b), and GQA (c). The middle row shows averaged attention maps, and the bottom row overlays them on the corresponding images. These visualizations are descriptive; they do not identify causal features or isolate the contribution of the Perceiver bottleneck.

4.4 Qualitative Analysis: The Perceiver Bottleneck

Figure 5 visualizes Perceiver cross-attention for four support examples from each of three tasks: Age prediction, emotion recognition, and GQA. The displayed maps are averaged across latent queries and attention heads. They show spatial variation in the visualization of support-image attention, but the patterns are diffuse and vary across examples. We do not interpret individual high-attention regions as identified facial features, emotion cues, or objects, nor do these maps establish that the model suppresses irrelevant information. In particular, averaged attention weights are descriptive rather than a measure of feature importance. The figure illustrates the behavior of the resampler on these examples; it does not establish that the Perceiver bottleneck causes the performance gains in Section 4. Isolating that contribution would require comparisons with alternative support aggregation mechanisms.

4.5 Is the gain just extra meta-training?

Because CoPA trains ϕ on labeled episodes, a fairer question than "CoPA vs. frozen ICL" is how much of the lift depends on in-domain supervision of ϕ. Table 3 (Appendix G) reports five training mixtures for the same architecture, evaluated at $K = 16$: mismatched specialists, intra-modality

generalists, inter-modality generalists, a universal mixture of all 14 datasets, and an in-domain oracle.

Three facts follow without new runs. First, in-domain oracles remain best (Age MAE 5.24, CIFAR-10 92.66%, OK-VQA 54.69%), so architecture plus target-domain meta-training is the upper envelope of this paper. Second, intra-modality mixtures transfer positively (CIFAR-10 87.10%, OK-VQA 40.53%), whereas inter-modality mixtures and some specialists go *below* the CIFAR-10 zero-shot baseline of 85.20% (VQAv2 specialist 78.70%, inter-modality 78.50%). Third, the universal mixture recovers much of the oracle on CIFAR-10 (90.80%) and improves OK-VQA (42.80%) without matching the VQA oracle (54.69%). CoPA can share structure within a task family and can context-switch when the mixture is diverse, but it can also negatively transfer. This is additional supervised training, not a free test-time property of the frozen VLM.

4.6 Computational Efficiency and Bottleneck Analysis

A fundamental limitation of visual in-context learning is the massive memory footprint of high-dimensional image tokens. To evaluate hardware efficiency, we profiled memory allocation and inference latency on a single A100 (80GB) GPU using Gemma-3. We decompose adaptation into Phase 1 (Context Adaptation) and Phase 2 (Query Prefill and Decode).

Table 2: **Computational Cost Analysis.** Profiling on an A100 (80GB). Metrics are presented as **Memory (GB) / Latency (ms)**. Standard ICL pushes the bottleneck to the Query phase, while gradient-based TTT pushes it to the Adaptation phase (triggering OOM at $K = 64$). CoPA stabilizes the Query phase to a constant $\mathcal{O}(1)$ complexity.

Method	**Phase (GB / ms)**	$K = 1$	$K = 4$	$K = 8$	$K = 16$	$K = 64$
Standard ICL	1: Adaptation	-	-	-	-	-
	2: Query Prefill	8.23 / 151	8.56 / 300	8.98 / 507	9.83 / 972	14.95 / 4542
Gradient TTT	1: Adaptation	10.42 / 266	17.28 / 650	26.50 / 1217	45.26 / 2552	**OOM**
	2: Query Prefill	8.20 / 120	8.20 / 147	8.20 / 120	8.20 / 120	-
CoPA (Ours)	1: Adaptation	8.55 / 172	8.98 / 396	9.56 / 745	10.72 / 1480	17.65 / 6765
	2: Query Prefill	**8.51 / 115**	**8.51 / 126**	**8.51 / 117**	**8.51 / 116**	**8.51 / 117**

Table 2 shows where each method pays for processing support examples. Frozen ICL retains them in the query prompt, so its reported prefill time rises from 151 ms at $K = 1$ to 4542 ms at $K = 64$. CoPA processes supports separately: its support-phase time rises from 172 to 6765 ms, while query prefill remains near 120 ms. For one query at $K = 16$, its measured phase times total 1596 ms, compared with 972 ms for frozen ICL; at $K = 64$, they total 6882 ms and 4542 ms, respectively. Thus CoPA reduces query-phase dependence on K, but these measurements do not show lower end-to-end latency for a single query. They also do not measure accuracy at $K = 64$.

5 Conclusion

CoPA maps a labeled visual support set to temporary low-rank updates for a frozen VLM. A Perceiver resampler compresses captured support activations, and an episodically meta-trained hypernetwork generates the updates without inference-time gradients. Across the evaluated tasks, CoPA outperforms the frozen ICL and one-step TTT implementations for $K \leq 16$. Its query-phase memory and prefill time remain approximately independent of support count in profiles through $K = 64$, although support processing becomes more expensive. These findings support amortized support-to-weight adaptation as a way to avoid retaining visual supports in the query prompt. The comparisons do not establish how much of the accuracy gain comes from the particular architecture rather than supervised meta-training. They also do not establish many-shot accuracy scaling or a single-query end-to-end latency advantage. Those questions remain open for supervision-matched baselines and more targeted component comparisons.

AI USE STATEMENT

We used generative AI tools to assist with implementing and editing software code, drafting and revising manuscript text, and improving its readability. We also used these tools to obtain feedback on the presentation of our methodology and experiments and on the interpretation of reported results, including the scope and limitations of our comparisons. The authors reviewed AI-assisted text, code, and suggestions, checked the manuscript against the underlying experiments, and take responsibility for the final methods, results, claims, and submitted content.

REPRODUCIBILITY STATEMENT

We provide code as supplementary material to reproduce the reported training, evaluation, baselines, and efficiency measurements. Section 3 specifies the support-to-weight procedure, and Section 4 describes the evaluated tasks and comparison protocols. The appendix documents the hypernetwork architecture, training setup, prompt formatting, dataset preparation, and implementation details. The supplementary code includes the configurations needed to run the reported experiments. Datasets and pretrained backbones must be obtained under their respective distribution terms. If the paper is accepted, we will also release the code through a public GitHub repository.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966, 2024.

Victor Akinwande, Mohammad Sadegh Norouzzadeh, Devin Willmott, Anna Bair, Madan Ravi Ganesh, and J Zico Kolter. Hyperclip: Adapting vision-language models with hypernetworks. *arXiv preprint arXiv:2412.16777*, 2024.

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. In *Forty-second International Conference on Machine Learning*.

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. In *International Conference on Machine Learning*, pp. 942–963. PMLR, 2025.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Léon Bottou and Vladimir Vapnik. Local learning algorithms. *Neural computation*, 4(6):888–900, 1992.

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.

Rujikorn Charakorn, Edoardo Cetin, Yujin Tang, and Robert Tjarko Lange. Text-to-lora: Instant transformer adaption. In *International Conference on Machine Learning*, pp. 7485–7514. PMLR, 2025.

Jieneng Chen, Luoxin Ye, Ju He, Zhao-Yang Wang, Daniel Khashabi, and Alan Yuille. Efficient large multi-modal models via visual context compression. *Advances in neural information processing systems*, 37:73986–74007, 2024a.

Yaofo Chen, Shuaicheng Niu, Yaowei Wang, Shoukai Xu, Hengjie Song, and Mingkui Tan. Towards robust and efficient cloud-edge elastic model adaptation via selective entropy distillation. *arXiv preprint arXiv:2402.17316*, 2024b.

Jiale Cheng, Yusen Liu, Xinyu Zhang, Yulin Fei, Wenyi Hong, Ruiliang Lyu, Weihan Wang, Zhe Su, Xiaotao Gu, Xiao Liu, et al. Glyph: Scaling context windows via visual-text compression. *arXiv preprint arXiv:2510.17800*, 2025.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. Learning how hard to think: Input-adaptive allocation of lm computation. *arXiv preprint arXiv:2410.04707*, 2024.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Haoyu Dong, Pengkun Zhang, Mingzhe Lu, Yanzhen Shen, and Guolin Ke. Machinelearninglm: Scaling many-shot in-context learning via continued pretraining. *arXiv preprint arXiv:2509.06806*, 2025.

Xinqi Fan, Xueli Chen, Luoxiao Yang, Chuin Hong Yap, Rizwan Qureshi, Qi Dou, Moi Hoon Yap, and Mubarak Shah. Test-time retrieval-augmented adaptation for vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8810–8819, 2025.

Honghao Fu, Yuan Ouyang, Kai-Wei Chang, Yiwei Wang, Zi Huang, and Yujun Cai. Contextnav: Towards agentic multimodal in-context learning. *arXiv preprint arXiv:2510.04560*, 2025.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Vlad Hosu, Hanhe Lin, Tamas Sziranyi, and Dietmar Saupe. Koniq-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE Transactions on Image Processing*, 29:4041–4056, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Jinwu Hu, Zhitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-time learning for large language models. *arXiv preprint arXiv:2505.20633*, 2025.

Shell Xu Hu, Pablo G Moreno, Yang Xiao, Xi Shen, Guillaume Obozinski, Neil D Lawrence, and Andreas Damianou. Empirical bayes transductive meta-learning with synthetic gradients. *arXiv preprint arXiv:2004.12696*, 2020.

Brandon Huang, Chancharik Mitra, Assaf Arbelle, Leonid Karlinsky, Trevor Darrell, and Roei Herzig. Multimodal task vectors enable many-shot multimodal in-context learning. *Advances in Neural Information Processing Systems*, 37:22124–22153, 2024.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Yixing Jiang, Jeremy Irvin, Ji Hun Wang, Muhammad Ahmed Chaudhry, Jonathan H Chen, and Andrew Y Ng. Many-shot in-context learning in multimodal foundation models. *arXiv preprint arXiv:2405.09798*, 2024.

Thorsten Joachims et al. Transductive inference for text classification using support vector machines. In *Icml*, volume 99, pp. 200–209, 1999.

Yuto Kojima, Jiarui Xu, Xueyan Zou, and Xiaolong Wang. Lora-ttt: Low-rank test-time training for vision-language models. *arXiv preprint arXiv:2502.02069*, 2025.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and better understanding vision-language models: insights and future directions. *arXiv preprint arXiv:2408.12637*, 2024.

Eugene Lee, Evan Chen, and Chen-Yi Lee. Meta-rppg: Remote heart rate estimation using a transductive meta-learner. In *European conference on computer vision*, pp. 392–409. Springer, 2020.

Mukai Li, Shansan Gong, Jiangtao Feng, Yiheng Xu, Jun Zhang, Zhiyong Wu, and Lingpeng Kong. In-context learning with many demonstration examples. *arXiv preprint arXiv:2302.04931*, 2023.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.

Lingyu Liang, Luojun Lin, Lianwen Jin, Duorui Xie, and Mengru Li. Scut-fbp5500: A diverse benchmark dataset for multi-paradigm facial beauty prediction. In *2018 24th International conference on pattern recognition (ICPR)*, pp. 1598–1603. IEEE, 2018.

Hanhe Lin, Vlad Hosu, and Dietmar Saupe. Kadid-10k: A large-scale artificially distorted iqa database. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–3. IEEE, 2019.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. *arXiv preprint arXiv:1805.10002*, 2018.

Yewei Liu, Xiyuan Wang, Yansheng Mao, Yoav Gelbery, Haggai Maron, and Muhan Zhang. Shine: A scalable in-context hypernetwork for mapping context to lora in a single pass. *arXiv preprint arXiv:2602.06358*, 2026.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, 2022.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pp. 3195–3204, 2019.

R Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D Hardy, and Thomas L Griffiths. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41):e2322420121, 2024.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. In *Proceedings of the 2022 conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2791–2809, 2022.

Naila Murray, Luca Marchesotti, and Florent Perronnin. Ava: A large-scale database for aesthetic visual analysis. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 2408–2415. IEEE, 2012.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.

Yihua Shao, Minxi Yan, Yang Liu, Siyu Chen, Wenjie Chen, Xinwei Long, Ziyang Yan, Lei Li, Chenyu Zhang, Nicu Sebe, et al. In-context meta lora generation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 6138–6146, 2025.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling test-time compute optimally can be more effective than scaling llm parameters. In *The Thirteenth International Conference on Learning Representations*, volume 18, 2025.

Yan Tai, Weichen Fan, Zhao Zhang, and Ziwei Liu. Link-context learning for multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27176–27185, 2024.

Arnuv Tandon, Karan Dalal, Xinhao Li, Daniel Koceja, Marcel Rød, Sam Buchanan, Xiaolong Wang, Jure Leskovec, Sanmi Koyejo, Tatsunori Hashimoto, et al. End-to-end test-time training for long context. *arXiv preprint arXiv:2512.23675*, 2025.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2023.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. Association for Computational Linguistics, 2024.

Zheng Xiong, Kang Li, Zilin Wang, Matthew Jackson, Jakob Foerster, and Shimon Whiteson. Hypervla: Efficient inference in vision-language-action models via hypernetworks. *arXiv preprint arXiv:2510.04898*, 2025.

Jingyuan Yang, Qirui Huang, Tingting Ding, Dani Lischinski, Danny Cohen-Or, and Hui Huang. Emoset: A large-scale visual emotion dataset with rich attributes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20383–20394, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Peiwen Yuan, Shaoxiong Feng, Yiwei Li, Xinglin Wang, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Heda Wang, Yao Hu, and Kan Li. Focused large language models are stable many-shot learners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6247–6261, 2024.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T Freeman, and Hao Tan. Test-time training done right. *arXiv preprint arXiv:2505.23884*, 2025.

Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

Yucheng Zhou, Xiang Li, Qianning Wang, and Jianbing Shen. Visual in-context learning for large vision-language models. *arXiv preprint arXiv:2402.11574*, 2024.

A Appendix

You may include other additional sections here.

B HyperNetwork Architecture and Initialization

This section details the exact mathematical formulation and hyperparameters of the Layer-Aware Residual Decoder and Perceiver Resampler used to generate the fast weights. Based on our implementation, the network operates primarily in a compressed hidden dimension ($d_{\text{model}} = 768$).

1. Context Adapter and Perceiver Resampler: Variable-length fused input activations $C^{(\ell)} \in \mathbb{R}^{B\times T\times d_{\text{in}}}$ (where $d_{\text{in}} = 2560$) are first projected to a fixed width d_{model} and normalized. To obtain a fixed-size representation, we utilize a Perceiver-style resampler with $M = 128$ learned latents. The resampler consists of $L_p = 6$ layers. Each layer applies cross-attention from the latents to the context, followed by latent self-attention, utilizing 12 attention heads, a feed-forward expansion factor of 4, and a dropout rate of 0.1 across all attention and feed-forward sub-layers. This compresses the context into task tokens $Z^{(\ell)} \in \mathbb{R}^{B\times M\times d_{\text{model}}}$.

2. Layer Identity Encoder: We compute a layer-identity embedding directly from the frozen weight matrix $W^{(\ell)} \in \mathbb{R}^{d_{\text{out}}\times d_{\text{in}}}$. We apply a linear compressor (no bias), mean-pool across the sequence dimension, and apply a final LayerNorm to yield a compact 1D identity vector $e^{(\ell)} \in \mathbb{R}^{1\times d_{\text{model}}}$.

3. Layer-Aware Residual Decoder: Given task tokens $Z^{(\ell)}$ and identity vector $e^{(\ell)}$, we decode a single query token into fast weights. We initialize a learned query vector $q \in \mathbb{R}^{1\times d_{\text{model}}}$ and expand it across batch episodes. We first condition the query on the target layer's identity, and then on the task tokens using multi-head cross-attention (8 heads):

$$q' = \mathrm{LN}\Big(q + \mathrm{Attn}(q, e^{(\ell)}, e^{(\ell)})\Big) \tag{3}$$

$$q'' = \mathrm{LN}\Big(q' + \mathrm{Attn}(q', Z^{(\ell)}, Z^{(\ell)})\Big) \tag{4}$$

We refine q'' using $L_d = 3$ residual MLP blocks with GELU activations. The refined vector is passed through linear projection heads to generate the final LoRA matrices and the scalar gate:

$$\begin{aligned} A^{(\ell)} &= \mathrm{reshape}(W_A q'') \in \mathbb{R}^{B\times d_{\text{in}}\times r}, \\ B^{(\ell)} &= \mathrm{reshape}(W_B q'') \in \mathbb{R}^{B\times r\times d_{\text{out}}}, \\ s^{(\ell)} &= 4.0 \cdot \sigma(w_s^\top q''). \end{aligned} \tag{5}$$

4. Stabilizing Initialization: To prevent the hypernetwork from destabilizing the frozen backbone early in training, we employ an identity-preserving initialization scheme. The projection matrix for the B-head (W_B) is initialized to exactly zero. The A-head (W_A) is initialized using Kaiming Normal ($a = 0.01$). The scalar gate bias is initialized to -5.0 before the sigmoid activation. This guarantees that $\Delta W^{(\ell)} \approx 0$ at initialization, meaning the adapted model initially behaves exactly like the frozen zero-shot model.

C Efficient Batch Training via Hyper-Linear Layers

A significant engineering challenge in episodic meta-training for ICL is that every sample in a training batch corresponds to a distinct task episode, requiring a distinct set of generated weights $\{A_i, B_i\}_{i=1}^{B}$. Standard deep learning frameworks are optimized for static weights shared uniformly across a batch dimension.

To overcome this bottleneck and maximize GPU utilization without relying on sequential `for`-loops, we implemented a custom `HyperBatchLinear` module. Instead of standard linear projections, we utilize optimized batched matrix multiplication (BMM). For a batch of input activations $X \in \mathbb{R}^{B\times T\times d_{\text{in}}}$, frozen weights W_0, and batch-specific generated adapters $A \in \mathbb{R}^{B\times d_{\text{in}}\times r}$ and $B \in \mathbb{R}^{B\times r\times d_{\text{out}}}$, the forward pass is computed in parallel as:

$$Y = XW_0^\top + s \cdot \Big((X \otimes A) \otimes B\Big) \tag{6}$$

where $\otimes$ denotes BMM. This ensures that memory and compute scale efficiently during meta-training across highly diverse task batches.

D Comprehensive Complexity Derivation

In this section, we provide a step-by-step formal derivation of the memory and computational complexity of our proposed CoPA framework compared to standard In-Context Learning (ICL).

Variables and Notation:

- K: Number of support shots.
- N_{shot}: Number of visual and textual tokens per shot.
- N_{ctx}: Total context length, where $N_{ctx} = K \cdot N_{shot}$.
- N_q: Number of tokens in the query prompt and generated response.
- d: Hidden dimension of the VLM.
- L: Total number of transformer layers.
- M: Number of latent tokens in the Perceiver resampler ($M \ll N_{ctx}$).
- r: Rank of the injected LoRA adapters ($r \ll d$).

D.1 Standard In-Context Learning (ICL)

In standard ICL, the entire context $\mathcal{S}$ must be concatenated with the query and processed by the frozen VLM.

Prefill Phase (Compute): The model processes the entire prompt of length $N_{ctx} + N_q$. The self-attention mechanism computes pairwise scores across all tokens, scaling quadratically with the sequence length. Additionally, the Q, K, V, and FFN projections scale with d^2. The prefill compute complexity per layer is:

$$\mathcal{O}((N_{ctx} + N_q)^2 \cdot d + (N_{ctx} + N_q) \cdot d^2)$$

Decoding Phase (Compute per token): For every new token generated, the model must attend to all previous tokens in the KV-cache. The compute cost per generated token per layer is:

$$\mathcal{O}((N_{ctx} + N_q) \cdot d + d^2)$$

This demonstrates that standard ICL inference slows down linearly as K increases.

Memory Footprint (KV-Cache): To avoid recomputing keys and values, the model caches them for all layers. The peak memory footprint required to generate the query response is:

$$\mathcal{O}((N_{ctx} + N_q) \cdot L \cdot d)$$

This linear scaling with respect to K is the primary cause of Out-of-Memory (OOM) errors in many-shot regimes.

D.2 CoPA (Ours)

CoPA decouples the support set processing (Encoding Phase) from the target inference (Query Phase).

Phase 1: Encoding (One-Time Cost) During the support forward pass, the model extracts intermediate activations $C^{(\ell)}$. While the frozen VLM still performs a forward pass over N_{ctx}, it is strictly under `no_grad`, and the KV-cache is *not* retained for the query phase. The Perceiver then cross-attends from M latent tokens to the captured support activations. Its cross-attention cost is linear in the support token count for fixed M, apart from model-width factors. This does not remove the compute cost of the preceding frozen VLM support pass. CoPA does not retain the support KV cache for subsequent query generation, although processing a longer support prompt can increase peak memory during the support phase.

Phase 2: Query Generation (Compute per token) Once the adapters $\Delta W^{(\ell)} = B^{(\ell)}A^{(\ell)}$ are injected, the support set is discarded. The model now only processes the N_q query tokens. The modified linear layers incur a negligible additional compute cost of $\mathcal{O}(r \cdot d)$ due to the low-rank matrix multiplications. The compute cost per generated token per layer becomes:

$$\mathcal{O}(N_q \cdot d + d^2 + r \cdot d)$$

Crucially, this is completely independent of N_{ctx}.

Phase 2: Query Memory Footprint Because the context N_{ctx} has been compressed into the model weights, the KV-cache only stores the N_q tokens of the query. The inference memory complexity is strictly:

$$\mathcal{O}(N_q \cdot L \cdot d)$$

Conclusion: With respect to the number of shots K, standard ICL exhibits $\mathcal{O}(K)$ memory scaling and $\mathcal{O}(K)$ compute scaling per generated token. In contrast, CoPA achieves $\mathcal{O}(1)$ memory scaling and $\mathcal{O}(1)$ compute scaling during the critical autoregressive query generation phase.

E Dataset Preparation Details

To ensure reproducibility and standardize the evaluation framework across 14 diverse datasets, we implemented a unified preprocessing pipeline. The datasets fall into two primary processing paradigms: standard vision tasks (regression and classification) and Visual Question Answering (VQA).

E.1 Regression and Classification Datasets

For fine-grained regression and few-shot classification benchmarks, labels were extracted either directly from standardized filenames (e.g., UTKFace, CIFAR10, MNIST, EmoSet, Food101, Stanford Cars) or aggregated from official metadata files:

- **Aesthetic and Beauty Scoring:** For AVA and SCUT-FBP5500, we aggregated the multiple human-annotated ratings per image and computed the mean score to serve as the continuous ground-truth target.
- **Image Quality Assessment:** For KADID-10k and KonIQ-10k, we mapped the provided Mean Opinion Scores (MOS) or Differential MOS (DMOS) directly from the official dataset distributions.

For datasets lacking official canonical splits in our raw downloads, we applied a deterministic *80/20* train-test split (enforced via a fixed random seed of 42) to partition the data into the support pool (training) and query evaluation sets (testing).

E.2 Visual Question Answering (VQA) Datasets

VQA tasks require mapping complex JSON-based question and annotation schemas to their corresponding images. We strictly utilized the official canonical splits to ensure comparability with prior literature and prevent data leakage:

- **VQAv2 and OK-VQA:** Both datasets were mapped to the MSCOCO image database, utilizing the `train2014` split for the support pool and the `val2014` split for testing.
- **GQA:** We utilized the `train_balanced` and `val_balanced` questions to ensure a uniform distribution of relational logic concepts.
- **TextVQA:** We mapped the official TextVQA v0.5.1 train and validation splits to the image repository.

For datasets providing multiple human annotations per question (VQAv2, OK-VQA, TextVQA), all valid answers were consolidated into a list to allow for robust target matching, whereas GQA provided a single canonical answer.

E.3 Offline Embedding Generation for kNN Retrieval

To support the Retrieval-Augmented kNN Selection baseline and CoPA (kNN) experiments discussed in the main text, we pre-computed visual embeddings for all images.

We utilized the `google/siglip-base-patch16-224` vision encoder to extract the CLS token embeddings (dimension of 768) for every image across all train and test splits. These embeddings were L2-normalized and cached offline. This caching strategy ensures that the retrieval mechanism evaluates solely cosine similarity without introducing vision-encoder bottlenecks during inference, allowing for highly efficient retrieval of the K most visually similar support examples for any given query.

F Training Setup and Task Formatting

To standardize the episodic meta-optimization across diverse multimodal tasks, we cast all continuous and discrete tasks into a unified language modeling format. The VLM is supervised via teacher-forced next-token prediction.

All models were trained using the AdamW optimizer with a learning rate of 5×10^{-5}, a batch size of 4 per device, and gradient accumulation steps of 8 for up to 2,000 steps which amounts to approximately 4 hours.

Prompt Templates:

- **Regression Tasks (e.g., Age Prediction, Quality Assessment):** Continuous scalars are rounded to canonical numeric strings. *Template:* `<|image|> What is the age of the person in the last image? Only output a number. [Label]`
- **Classification Tasks (e.g., CIFAR10, EmoSet):** The category list is explicitly provided in the query to align with standard ICL evaluation protocols.
  Template: `Question: What you see in this image? There is a category list: [class_1, class_2, ...]. Image {i}: <|image|> . Answer: [Label]`
- **Visual Question Answering (e.g., VQAv2, GQA):**
  Template: `<|image|> Answer the following question using a single word or a short phrase. Question: {q} Answer: [Label]`

During training, cross-entropy loss is strictly masked (`label = -100`) for all prompt text and image tokens; the loss is computed exclusively on the text tokens comprising the ground-truth completion.

Implementation and Precision Constraints: To ensure memory efficiency and stabilize training, all frozen backbone models, fused activations, and HyperNetwork operations were strictly cast to `bfloat16` mixed precision. Furthermore, during the dynamic injection phase, we enforced a strict architectural boundary: the targeted LoRA adapters were isolated exclusively to the language model and cross-modal projection pathways. The foundational vision tower was completely excluded from any parameter adaptation to preserve the integrity of the pre-trained visual representations.

G Out-of-Distribution (OOD) Transferability and Task Diversity

We examine how the source and diversity of episodic meta-training data affect CoPA on Age, CIFAR-10, and OK-VQA. Some configurations exclude the target dataset during meta-training; the all-task and target-domain configurations include it. We therefore report these conditions separately rather than treating every row as transfer to an unseen task.

We systematically ablate the diversity of the meta-training mixture to observe how adaptation performance scales. The configurations are defined as follows:

1. **Specialist (Single Mismatched Task):** Trained exclusively on one highly specific, out-of-domain task.
2. **Generalist (Intra-Modality):** Trained on multiple datasets strictly within the target's modality, but excluding the target dataset itself (e.g., tested on CIFAR10, trained on EmoSet, Food101, MNIST, Stanford Cars).
3. **Generalist (Inter-Modality):** Trained on a diverse mixture of datasets from the *other two* modalities, completely withholding the target's modality (e.g., tested on Classification, trained strictly on Regression and VQA datasets).
4. **Universal (All Tasks):** A single set of HyperNetwork weights trained simultaneously on all 14 datasets, including the target. This tests the model's ability to dynamically "context switch" purely based on the inference support set.
5. **Oracle (In-Domain):** Trained *only* on the target dataset (a target-specific specialist).

Table 3: **Transferability at $K = 16$.** Age MAE ↓, CIFAR-10 accuracy ↑, OK-VQA accuracy ↑. Intra-modality mixtures transfer positively; some specialists and inter-modality mixtures fall below the CIFAR-10 zero-shot baseline. The universal mixture approaches the oracle on CIFAR-10 but does not match the VQA oracle.

Model Type	Training Source	Age (MAE ↓)	CIFAR10 (Acc ↑)	OK-VQA (Acc ↑)
Baseline	Zero-Shot (No Adaptation)	7.83	85.20	34.83
Specialist	Aesthetic (Regression Only)	<u>5.59</u> ↑	80.00 ↓	35.30 ↑
	EmoSet (Classification Only)	6.04 ↑	<u>87.30</u> ↑	36.27 ↑
	VQAv2 (VQA Only)	**5.52** ↑	78.70 ↓	**45.47** ↑
Generalist	**Intra-Modality** (Same Task Type)	6.19 ↑	87.10 ↑	40.53 ↑
	Inter-Modality (Cross-Task Types)	6.81 ↑	78.50 ↓	37.50 ↑
Universal	**All Tasks** (Context Switching)	5.82 ↑	**90.80** ↑	<u>42.80</u> ↑
Oracle	*Target Domain Only*	*5.24*	*92.66*	*54.69*

Analysis of Transferability and Context Switching Dynamics: Table 3 highlights several critical dynamics regarding how HyperNetworks internalize in-context learning:

- **Modality Mismatch and Negative Transfer:** HyperNetworks trained on extremely narrow domains or across fundamentally misaligned modalities suffer from negative transfer.

For example, the Aesthetic specialist, the VQAv2 specialist, and the Inter-Modality generalist all degrade CIFAR10 performance below the 85.20% zero-shot baseline (dropping to 80.00%, 78.70%, and 78.50%, respectively). Without sufficient task diversity within the target modality, the network overfits its parameter generation strategy, forcing irrelevant logic (e.g., spatial coordinate regression or autoregressive text generation) onto the target domain.

- **Intra-Modality Synergy vs. Inter-Modality Interference:** The **Intra-Modality** generalist consistently yields positive transfer across all benchmarks (e.g., improving OK-VQA to 40.53% and CIFAR10 to 87.10%). This proves CoPA effectively learns shared structural priors within a specific task family. Conversely, the struggles of the **Inter-Modality** model highlight the difficulty of adapting to an entirely unseen output manifold without representative anchor tasks in the meta-training mixture.
- **Asymmetric Transfer Across Task Types:** The results reveal a notable asymmetry in task transferability. Simple continuous regression targets (Age Prediction) are highly receptive to features learned from more complex modalities. Notably, the VQAv2 specialist achieves a remarkable 5.52 MAE on Age Prediction, vastly outperforming the zero-shot baseline (7.83) and rivaling the Oracle (5.24). This suggests that the dense, multi-grained visual reasoning required for VQA yields robust, generalized representations that naturally benefit regression tasks. **All-task mixture.** Training on all 14 datasets, including each evaluated target, yields an Age MAE of 5.82, CIFAR-10 accuracy of 90.80%, and OK-VQA accuracy of 42.80%. These results show that a single trained model can perform on several evaluated datasets. They do not establish that it learns a general matching algorithm or transfers to a target omitted from training.

H Adaptation Sites and Module Targeting

The selection of the adaptation site is a critical design choice that determines the balance between parameter efficiency and task-specific expressivity. We conduct a systematic ablation across layer depths, distribution strategies, and specific transformer components using **Gemma 3 4B-it** as the backbone.

Table 4: **Ablation on Injection Sites (Gemma 3 4B-it).** Performance evaluated on UTKFace (MAE ↓) and EmoSet (Acc ↑) across $K = 16$ shots. Best results per section are **bolded**, second best are underlined.

Placement Strategy	Layer Indices	Age (MAE) ↓	EmoSet (Acc) ↑
Depth Analysis (Single Layer)			
Vision-Language Bridge	{0}	5.18	71.0
Shallow	{4}	**5.11**	<u>72.2</u>
Mid (Main Paper)	**{8}**	<u>5.15</u>	**74.7**
Deep	{32}	5.87	67.6
Distributed Strategies			
Sparse (Edges)	{0, 32}	5.31	74.0
Sparse (Distributed)	{4, 14, 24}	<u>5.26</u>	74.1
Mid-Cluster	{7, 8, 9}	**5.20**	<u>74.4</u>
All Layers	**{0, . . . , 32}**	12.35	**76.3**
Component Target (Index 8)			
Attention (q_proj only)	{8}	6.01	60.7
Attention (v_proj only)	{8}	5.73	65.3
FFN (up_proj only)	{8}	**5.04**	**78.1**
FFN ($down_proj$ only)	**{8}**	<u>5.21</u>	<u>76.4</u>

Analysis of Adaptation Dynamics: The results in Table 4 reveal several critical insights regarding how visual in-context learning manifests within the VLM architecture:

- **The Shallow-to-Mid Sweet Spot:** Contrary to the assumption that deeper layers exclusively handle complex reasoning, targeting the deepest layer (Layer 32) yields the poorest single-layer classification performance. This suggests late-stage representations are too rigidly linguistic to be effectively recalibrated by raw visual context. Instead, the model exhibits a "sweet spot" in the shallow-to-mid layers (Layers 4 and 8), where multimodal features are sufficiently processed but remain malleable.
- **The Modality Divergence of Dense Adaptation:** Adapting **All Layers** presents a striking trade-off depending on the task modality. For discrete semantic classification (EmoSet), global adaptation provides a significant performance boost (76.3%), suggesting that pervasive weight updates greatly aid deep semantic realignment. However, for fine-grained continuous regression (Age Prediction), this dense distribution over-parameterizes the HyperNetwork's output capacity, causing severe optimization instability and a catastrophic spike in error (MAE 12.35). Conversely, targeted **Mid-Clusters** (Layers 7, 8, 9) yield highly stable, competitive results across both modalities without the parameter explosion.
- **Component Sensitivity (FFN vs. Attention):** Adapting the Feed-Forward Network components (`up_proj`, `down_proj`) strictly dominates the Attention projections (`q_proj`, `v_proj`). This empirically validates the hypothesis that FFNs act as the primary key-value associative memory for task-specific logic, making them the optimal adaptation sites for test-time reconfiguration.

Choice used in the main paper. Main experiments adapt layer 8, `down_proj` only. On UTKFace and EmoSet this site is stable across modalities, but it is not the best single component: `up_proj` is stronger on both metrics. We retain `down_proj` as a frozen default across all datasets and backbones rather than retuning the site per task. Treating up-projection, or a sparse mid-layer cluster, as the primary site is left to future work.

I Performance Scaling Across K-Shots

A well-documented limitation of standard In-Context Learning (ICL) in Vision-Language Models is visual context overload: blindly appending demonstrations to the prompt often degrades performance or causes out-of-memory (OOM) failures due to attention dilution and massive KV-cache expansion. In Section 4.3 of the main text, we introduced this phenomenon conceptually. Here, we provide the explicit quantitative trajectories across varying numbers of shots (K) to demonstrate CoPA's stability.

Analysis of Scaling Trajectories: The scaling trajectories across Age Prediction, CIFAR10, and VQAv2 (Figure 6) highlight the distinct mechanistic advantage of our architecture:

- **The Pathology of Context Overload:** Standard ICL and TTT exhibit severe performance degradation as the number of shots increases. Across all three modalities, adding more examples initially harms performance, creating a U-shaped degradation curve for accuracy and an inverted U-shape for error. For instance, Standard ICL and TTT accuracy on CIFAR10 drops precipitously from 89.0% at $K = 1$ down to 78.0% at $K = 8$. Similarly, Age Prediction MAE spikes drastically above the 7.83 zero-shot baseline, peaking past 12.0 at $K = 8$. This perfectly illustrates how base attention mechanisms become distracted and overwhelmed by an inflating, noisy context window.
- **CoPA's Robust Stability:** By decoupling the support context from the query through targeted adaptation sites, CoPA entirely sidesteps context overload. It establishes a dominant performance margin immediately at $K = 1$ (e.g., jumping straight to $\sim$74% on VQAv2 compared to the $\sim$57% baseline) and maintains strict stability as the support set grows. Because the visual context is bottlenecked into a fixed, highly compressed budget of tokens ($M = 128$) regardless of the support set size, the downstream HyperNetwork never faces an overloaded context window and seamlessly aggregates statistical evidence.
- **Consistent Cross-Modality Scaling:** This robust scaling behavior is not isolated to a single type of reasoning. CoPA maintains its flat, stable trajectory whether the task requires continuous regression (Age), categorical classification (CIFAR10), or generative text an-

swering (VQAv2). This confirms that the targeted parameter generation successfully isolates task-specific semantics from context-length limitations.

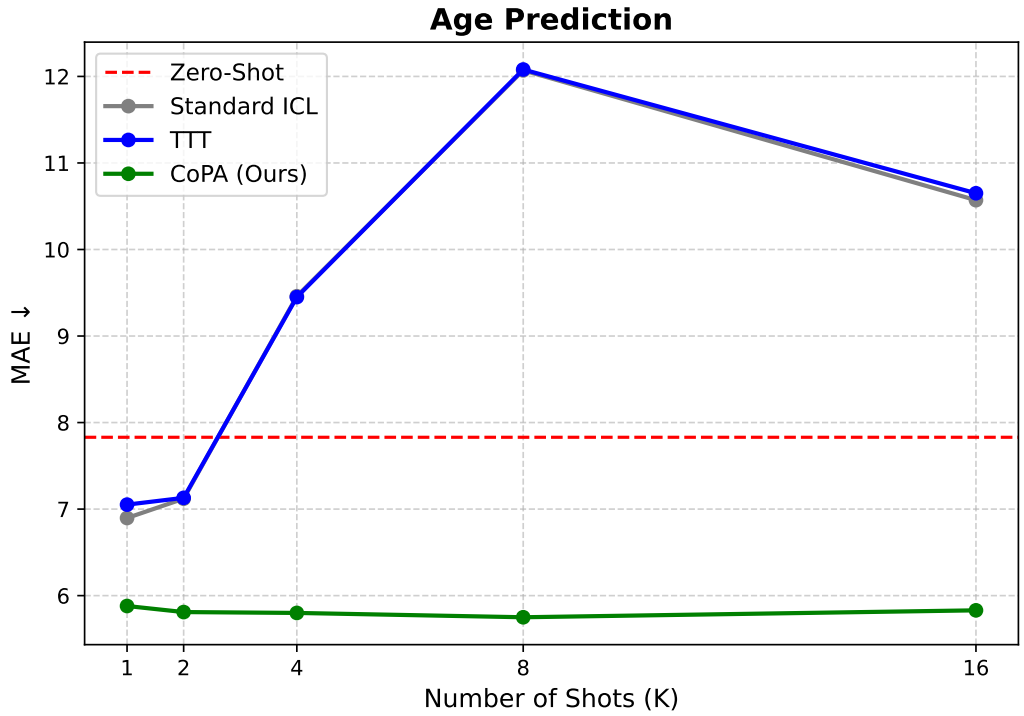
J VISUALIZATIONS OF THE PARAMETER MANIFOLD

To verify that CoPA learns structured, task-specific parameter manifolds, we visualize the t-SNE projections of the generated target weights ($\Delta\Theta$) produced by the HyperNetwork. We extract inference episodes from six diverse datasets across our three target modalities: Age Prediction, Aesthetic Scoring, EmoSet Classification, CIFAR-10, VQAv2, and GQA.

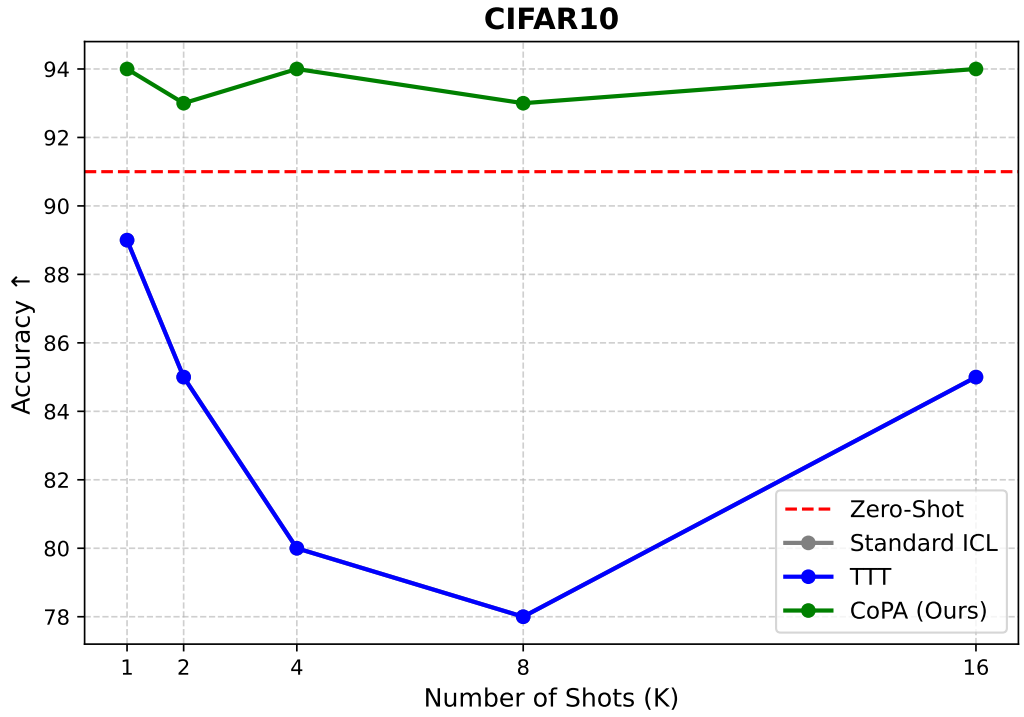
We compare the weight spaces generated by two distinct meta-training regimes: task-specific specialists (individual models trained on a single task) and the Universal model (a single shared checkpoint trained simultaneously on all tasks).

The visualizations reveal several key insights into CoPA's in-context adaptation mechanism:

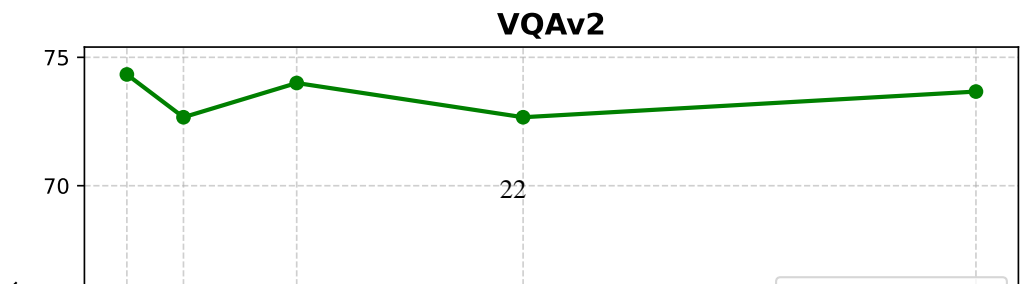
- **Task Isolation in Specialists:** As shown in Figure 7a, task-specific checkpoints produce perfectly isolated parameter clusters. Because these networks are trained independently, their resulting weight generation spaces are highly disjoint and do not need to negotiate a shared semantic topology.
- **Dynamic Context Switching:** The shared universal checkpoint (Figure 7b) successfully partitions its parameter space to accommodate all six tasks. Even though all target weights are generated by the exact same frozen HyperNetwork, the model dynamically routes the backbone into distinct, task-specialized manifolds purely based on the visual and textual semantics of the inference support set.
- **Modality-Aware Representation Sharing:** Crucially, the shared checkpoint naturally organizes the parameter space by task similarity. While structurally different tasks (like Age Prediction and CIFAR-10) remain widely separated, closely related datasets belonging to the same modality (such as VQAv2 and GQA) exhibit significant overlap. This indicates that the Universal HyperNetwork avoids naive task memorization; instead, it actively learns and reuses shared, generalized sub-algorithms for related reasoning paradigms.

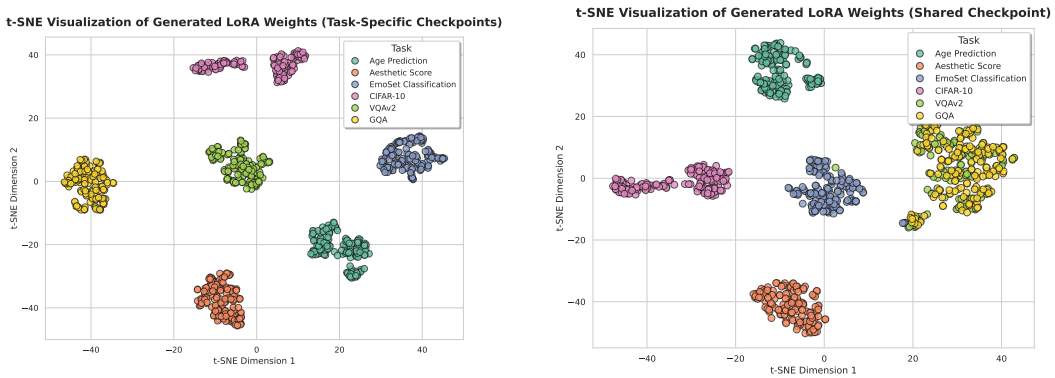


(a) Age prediction



(b) CIFAR-10 (Accuracy ↑)





(a) Task-Specific Checkpoints

(b) Shared Checkpoint (Universal)

Figure 7: **t-SNE Visualization of Generated Weights.** (a) Weights generated by isolated, task-specific models form highly disjoint, orthogonal clusters. (b) The Universal model dynamically context-switches, maintaining distinct clusters for different tasks while intrinsically mapping highly related tasks (e.g., VQAv2 and GQA) into overlapping regions of the parameter space, demonstrating shared representation learning.