Adaptive Weighted Proxy Tuning: Efficient Gray-Box Steering for Image Captioning

Nafew Azim[1] **Fuad Rahman**[2] **Nabeel Mohammed**[1]
[1]Department of Electrical and Computer Engineering, North South University
[2]Apurba Technologies
{nafew.azim, nabeel.mohammed}@northsouth.edu
fuad@apurbatech.com

Abstract

Adapting Large Vision-Language Models (LVLMs) to specialized domains typically demands resource-intensive fine-tuning or access to proprietary parameters ("white-box" access). While decoding-time strategies like Proxy Tuning offer a parameter-efficient alternative, they rely on rigid, static logit arithmetic that fails to account for instance-specific variations in model certainty and domain shift. In this work, we introduce **Adaptive Weighted Proxy Tuning (AWPT)**, a *gray-box* steering framework that dynamically modulates the logit contributions of a large base model, a fine-tuned expert, and an untuned anti-expert. Unlike static approaches, AWPT introduces two instance-aware mechanisms: (1) a lightweight **ViT-based Weight Predictor** that performs amortized inference to estimate optimal mixing coefficients in real-time with negligible added latency (∼0.03s overhead), and (2) a **Per-Sample Optimization** objective that establishes theoretical performance bounds via gradient-based logit steering. Extensive evaluation across medical (ROCOv2, IU-Xray) and general domains (Flickr30k, MS COCO, TextCaps) demonstrates that AWPT achieves performance parity with fully fine-tuned models while remaining parameter-free regarding the generator. Crucially, our dynamic weighting acts as an effective regularizer, significantly reducing object hallucinations; however, while AWPT provides a robust pathway for deploying general-purpose LVLMs, this technology is designed strictly as a human-in-the-loop assistive tool in safety-critical contexts.

Keywords: Image Captioning, Vision-Language Models (VLM), Logit Steering, Model Arithmetic.

1 Introduction

The adaptation of Large Vision-Language Models (LVLMs) to specialized domains—such as radiology or technical diagramming—presents a dilemma. While scaling laws dictate that larger models yield superior reasoning, fine-tuning these billion-parameter giants is often computationally prohibitive or infeasible due to proprietary restrictions (e.g., API-only access). Consequently, "gray-box" adaptation methods, which steer model behavior using only output logits without access to internal weights or gradients, have gained significant traction.

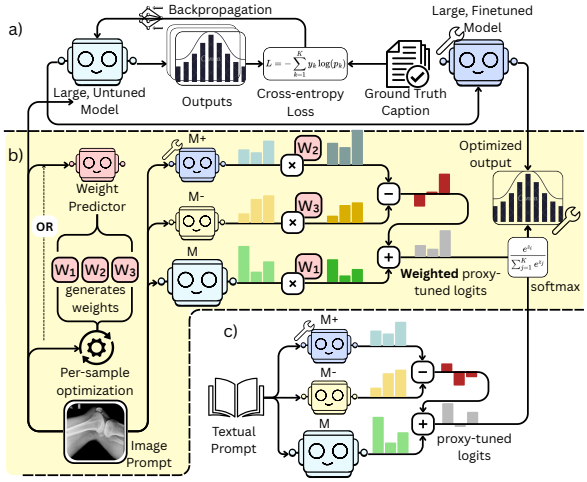

Figure 1: Contrast of adaptation paradigms. (a) **Traditional Fine-tuning** requires expensive parameter updates to the large model. (b) **Standard Proxy Tuning** applies a static, global correction using small expert/anti-expert models. (c) **Adaptive Weighted Proxy Tuning (Ours)** dynamically modulates the contribution of each model per image via learnable weights $\mathbf{w}$, allowing the system to rely on the expert for domain-specific samples (e.g., medical X-rays) while deferring to the base model for general scenes.

A promising direction in this space is *Proxy Tuning* (Liu et al., 2024), which steers a large base model by injecting the logit difference between a small fine-tuned "expert" and a small untuned "anti-expert." While effective for general language tasks, current proxy tuning approaches suffer from a critical limitation: they apply a static arithmetic adjustment across all inputs. This formulation as-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 1197–1217

sumes that the "expert" is equally reliable, and the "anti-expert" equally detrimental, for every instance. In reality, captioning difficulty is highly heterogeneous; a complex medical anomaly may require strong expert guidance, whereas a generic object scene is often better handled by the robust prior of the large base model.

To bridge this gap, we introduce **Adaptive Weighted Proxy Tuning (AWPT)**, a decoding-time framework that transforms static logit arithmetic into a dynamic, instance-aware steering mechanism. By modulating the contribution of the base, expert, and anti-expert models via learnable scalar weights, AWPT enables precise control over the generation process without modifying the large model's parameters.

We introduce two complementary strategies to estimate these weights:

1. **ViT-based Weight Predictor (Amortized Inference):** A lightweight module trained to regress optimal mixing coefficients from image features in a single forward pass. This method incurs negligible added latency ($\sim$0.03s) while outperforming static baselines, making it viable for real-time deployment.

2. **Per-Sample Optimization (PSO):** A test-time optimization procedure that iteratively refines weights to minimize cross-entropy against a target distribution in the logit space. While computationally heavier, this establishes the theoretical performance upper bound of the weighted proxy framework.

We rigorously evaluate AWPT across five datasets. Our primary evaluation focuses on four datasets, including the specialized medical domains of ROCOv2 and IU-Xray, and general benchmarks like Flickr30k and TextCaps, while our extended evaluation on MS COCO is detailed in Appendix A.4. Unlike prior work that focused solely on n-gram overlap metrics (BLEU/CIDEr), we explicitly measure semantic faithfulness (BERTScore) to ensure clinical precision. Our results demonstrate that AWPT not only matches the performance of fully fine-tuned models but significantly reduces object hallucinations compared to standard proxy tuning.

Our contributions are:

- We propose the first adaptive weighting framework for proxy tuning in image captioning, addressing the limitations of static logit arithmetic in multi-domain settings.

- We provide a "gray-box" solution that achieves parity with fine-tuned models on specialized tasks (e.g., radiology) without requiring parameter access to the large generator.

- We demonstrate that dynamic weighting acts as a robust regularizer, significantly reducing hallucination rates by suppressing generic "anti-expert" tokens more effectively than static baselines.

2 Related Work

2.1 Parameter-Efficient Transfer Learning (PETL)

While full fine-tuning remains the gold standard for adaptation, the exploding size of LVLMs has necessitated efficient alternatives. Techniques such as Adapters (Houlsby et al., 2019), Prefix Tuning (Li and Liang, 2021), and Low-Rank Adaptation (LoRA) (Hu et al., 2022; Dettmers et al., 2023) drastically reduce the number of trainable parameters. However, these "white-box" methods fundamentally require access to the model's internal weights and gradients, rendering them inapplicable to proprietary models accessed via APIs. Our work targets the "gray-box" setting (Liu et al., 2024), where one must adapt the generator solely through access to its output probability distribution, circumverting the need for weight inspection or modification.

2.2 Inference-Time Intervention and Steering

Recent research has pivoted toward steering model behavior at decoding time by manipulating output logits. **DExperts** (Liu et al., 2021) introduced the concept of "expert" and "anti-expert" logit subtraction to reduce toxicity. This paradigm has been extended to hallucination reduction in vision-language models: **Contrastive Decoding (CD)** (Li et al., 2023b) penalizes tokens favored by a weak amateur model to improve coherence; **Visual Contrastive Decoding (VCD)** (Leng et al., 2024) contrasts logits from distorted visual inputs to isolate and suppress object hallucinations. Similarly, **ADACAD** (Wang et al., 2024) dynamically balances contextual and parametric knowledge.

While effective, these methods typically rely on *heuristic* or *fixed* hyperparameters (e.g., a constant penalty weight) to govern the intervention

strength. Recent theoretical work has explored closed-form multi-objective decoding (Shi et al., 2024), but often assumes static trade-offs. Our Adaptive Weighted Proxy Tuning (AWPT) generalizes these approaches by treating the mixing coefficients as dynamic, learnable latent variables that adjust per-instance, allowing the system to modulate the intervention strength based on the complexity of the visual input.

2.3 Logit Arithmetic and Proxy Tuning

Building on DExperts, **Proxy Tuning** (Liu et al., 2024) formalized the framework of "tuning" a black-box base model (M_{base}) by injecting the logit difference between a small fine-tuned expert (M_{exp}) and a small untuned anti-expert (M_{anti}). This effectively transfers the domain shift captured by the small expert to the large base model. However, standard Proxy Tuning applies a static arithmetic operation: $\ell_{\text{final}} = \ell_{\text{base}} + \alpha(\ell_{\text{exp}} - \ell_{\text{anti}})$. This rigidity ignores the heterogeneity of data; complex medical anomalies may require aggressive expert intervention ($\alpha \gg 1$), while generic scenes may degrade if the small expert is over-weighted. By introducing *learnable, instance-aware weights* to this triad, our work bridges the gap between static logit arithmetic and full fine-tuning, offering a rigorous upper bound for decoding-time adaptation.

3 Methodology

3.1 Problem Formulation and Gray-Box Setting

We address the problem of steering a Large Vision-Language Model (LVLM), denoted as M_l, toward a target domain distribution without accessing or modifying its internal parameters θ_l. Let x be an input image and $y = (y_1, \ldots, y_T)$ be a target caption. At each decoding step t, the model produces a logit vector $\mathbf{z}_t \in \mathbb{R}^{|\mathcal{V}|}$ over the vocabulary $\mathcal{V}$.

We assume a **Gray-Box** setting: we have read-access to the output logits $\mathbf{z}_t$ of M_l, but write-access to parameters is forbidden. To guide generation, we employ two auxiliary models: a small fine-tuned "expert" (M_t) and a small untuned "anti-expert" (M_u).

Assumption 1 (Vocabulary Alignment): To perform valid logit arithmetic, we assume the vocabulary spaces of M_l, M_t, and M_u are aligned, or that a bijective mapping $\Phi : \mathcal{V}_{\text{small}} \to \mathcal{V}_{\text{large}}$ exists. In this work, we satisfy this by strictly selecting models from the same architectural family. Intra-family models (e.g., Qwen2.5-VL 3B and 7B variants) share the same tokenizer by design. This guarantees an exact vocabulary index mapping, which enables reference-free adaptive logit arithmetic directly across output distributions without the need for empirical alignment (see Appendix A.9).

3.2 Adaptive Weighted Proxy Tuning (AWPT)

Standard proxy tuning (Liu et al., 2024) injects a static domain shift into the base model: $\tilde{\mathbf{z}} = \mathbf{z}_l + \alpha(\mathbf{z}_t - \mathbf{z}_u)$. This assumes a constant "value of expertise" across all inputs. The overall architecture of Adaptive Weighted Proxy Tuning (AWPT), illustrating both the Amortized Inference and Per-Sample Optimization paths, is depicted in Figure 2. We relax this assumption by introducing a learnable weight vector $\mathbf{w}(x) = [w_l, w_f, w_u] \in \mathbb{R}^3_{\geq 0}$ that modulates the contribution of each model instance-wise. The modified logit $\tilde{\mathbf{z}}(x)$ is defined as:

$$\tilde{\mathbf{z}}(x) = w_l \cdot \mathbf{z}_l(x) + (w_f \cdot \mathbf{z}_t(x) - w_u \cdot \mathbf{z}_u(x)) \quad (1)$$

The decoding distribution is then $p(y_t|y_{<t}, x) = \text{Softmax}(\tilde{\mathbf{z}}(x))$. By dynamically adjusting $\mathbf{w}$, the system can aggressively suppress hallucinations ($w_u \uparrow$) or defer to the base model's general knowledge ($w_l \uparrow$) as required by the image complexity. Ablations on parameter expansion (Appendix A.3.2) confirm that while scalar weights provide sufficient steering, additional bias or global temperature scaling yields no semantic gains.

3.3 Strategy I: Amortized Inference via Weight Prediction

For real-time deployment, calculating optimal weights per-sample via optimization is cost-prohibitive. We therefore propose *Amortized Inference* (detailed in Algorithm 1, Appendix A.8), training a lightweight auxiliary network $\mathcal{W}_\phi$ to predict $\mathbf{w}$ directly from the image.

3.3.1 Architecture

The Weight Predictor $\mathcal{W}_\phi$ utilizes a frozen ViT-Base backbone (Dosovitskiy et al., 2020) to extract global visual features $v = \text{ViT}(x) \in \mathbb{R}^{768}$. A lightweight projection head (MLP) maps these features to the weight simplex:

$$\hat{\mathbf{w}} = \sigma(\text{MLP}(v)) \quad (2)$$

where σ is the Sigmoid function, bounding weights to $[0, 1]$. Note that we process the image independently of the text decoding steps, introducing a negligible latency overhead (approx. 20ms).

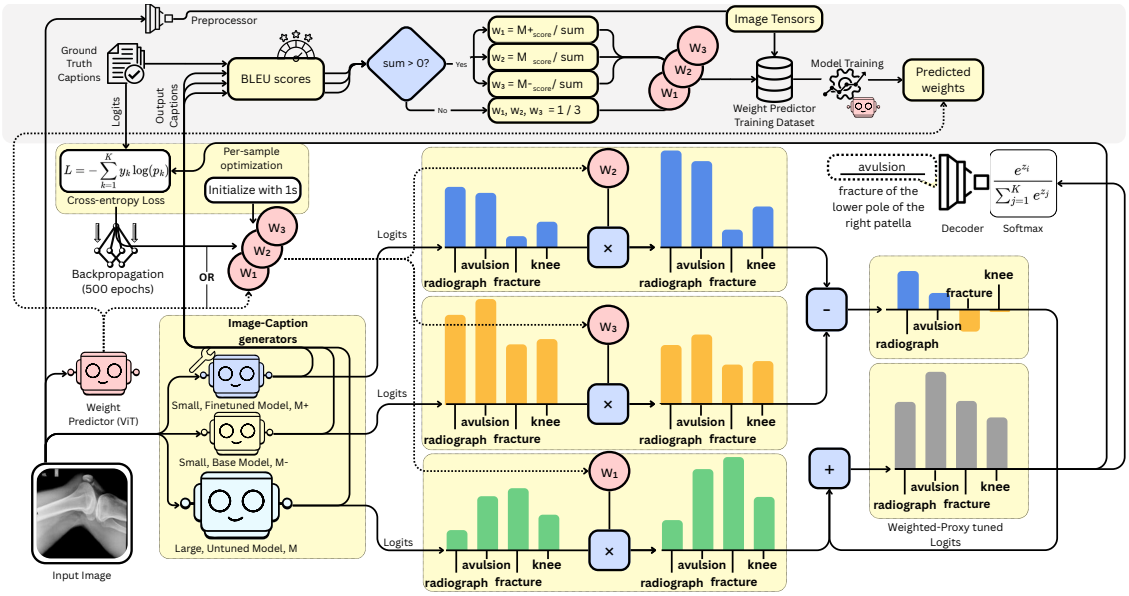

Figure 2: **Overview of Adaptive Weighted Proxy Tuning (AWPT).** The framework operates in a gray-box setting, requiring only logit access to the large base model (M_l). We modulate the influence of the base model, a fine-tuned expert (M_t), and an untuned anti-expert (M_u) using scalar weights $\mathbf{w} = [w_l, w_f, w_u]$. These weights are estimated via one of two paths: (Top/Gray) An amortized **Weight Predictor** (ViT-Base + MLP) that regresses optimal coefficients from the image in a single forward pass; or (Bottom) **Per-Sample Optimization**, which iteratively updates $\mathbf{w}$ via gradient descent on the logit space to establish performance upper bounds.

3.3.2 Oracle Supervision and Training

To train $\mathcal{W}_\phi$, we construct a dataset of "Oracle" weights $\mathbf{w}^*$. For a given training image x_i, we generate captions using M_l, M_t, and M_u independently and evaluate their quality using a reference-based metric (e.g., CIDEr), denoted $S(\cdot)$. The target weights are derived from the normalized relative performance:

$$\mathbf{w}^*_{i,k} = \frac{S(M_k(x_i))}{\sum_{j\in\{l,t,u\}} S(M_j(x_i))} \tag{3}$$

The predictor is trained to minimize the Mean Squared Error (MSE) between predicted $\hat{\mathbf{w}}$ and oracle $\mathbf{w}^*$. This effectively distills the "trustworthiness" of each component model into the predictor. This distillation aligns the predictor with mixture-aware oracles, capturing 95–98% of the theoretical performance bound established by Per-Sample Optimization (PSO) while enabling real-time inference at ∼0.018s per image (validated in Appendix A.7).

3.4 Strategy II: Instance-Optimal Steering (Per-Sample Optimization)

To establish the theoretical performance upper bound of the weighted proxy framework, we introduce a gradient-based optimization method. This strategy assumes access to a reference caption (or a strong teacher signal) to directly optimize $\mathbf{w}$ at inference time.

3.4.1 Differentiable Objective

Unlike prior works that attempt to optimize discrete decoding metrics (non-differentiable), we formulate the objective in the logit space. Let y be the target token sequence. We seek $\mathbf{w}$ that minimizes the Cross-Entropy (CE) loss of the *mixed* logits against the target:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in[0,1]^3} \mathcal{L}_{\text{CE}}\left(\text{Softmax}(\tilde{\mathbf{z}}(\mathbf{w})), y\right) \tag{4}$$

This formulation allows for backpropagation through the mixing operation directly to the scalar weights $\mathbf{w}$, without requiring gradients from the model parameters themselves (which remain frozen).

3.4.2 Optimization Procedure

We initialize $\mathbf{w} = [1, 1, 1]$. At each iteration, we perform a forward pass through the three models (using teacher forcing with y) to obtain the logit tensors. We then compute the mixed logits, calculate the loss, and update $\mathbf{w}$ using Adam. To respect the constraint $\mathbf{w} \in [0, 1]$, we apply projected gradient descent. This method, detailed in Algorithm 2 (Appendix A.8), typically converges in ∼500 iterations.

Crucially, Per-Sample Optimization (PSO) utilizes ground-truth reference captions specifically from the benchmark *test splits*. We emphasize that PSO is purely used to find the mathematically optimal steering weights to establish a theoretical performance upper bound. In contrast, our deployable network (the Weight Predictor) operates entirely reference-free at test time. While computationally expensive, PSO provides a rigorous "Skyline" for what AWPT can achieve under ideal weighting conditions. Our amortized predictor matches the rigorous PSO bound ($\sim$0.667s overhead) while requiring only $\sim$0.018s of added computation, enabling real-time deployment.

4 Experiments

4.1 Experimental Setup

For rigorous evaluation of Weighted Proxy Tuning (WPT), we fine-tuned ten vision–language backbones across four captioning benchmarks: BLIP-Base (Li et al., 2022), BLIP-2-Base (Li et al., 2023a), CLIP-Base (GPT-2 Decoder) (Mokady et al., 2021), Florence-2-Base (Xiao et al., 2023), Qwen2.5-VL-3B-Instruct (Qwen Team, 2025), SmolVLM-500M-Instruct (Allal et al., 2025), Gemma-3-Base (Gemma Team, 2025), **InternVL-3** (Zhu et al., 2025), **LLaVA-NeXT** (Li et al., 2024), and **PaliGemma** (Beyer et al., 2024) (the last three as high-performance references).

Training Protocol: All models were fine-tuned using distributed training on a mixed NVIDIA GPU cluster (comprising RTX 4090, RTX 5090, L40s, and RTX 4060 Ti). To maintain parameter efficiency and ensure fair comparison with our proxy methods, we utilized Low-Rank Adaptation (LoRA) with rank $r = 16$ and $\alpha = 16$. Optimization was performed via AdamW in 8-bit precision (learning rate 2×10^{-5}, weight decay 0.01) with a linear warmup of 5 steps. We employed an effective batch size of 8 per GPU (per-device batch=2, gradient accumulation=4). All images were resized to 224×224 and normalized using standard ImageNet statistics. See Appendix A.5 for Weight Predictor training details.

Checkpoint Selection: Addressing concerns regarding fixed-epoch comparisons, we eschewed arbitrary stopping criteria. Instead, we monitored CIDEr scores on the validation set and selected the best-performing checkpoint for each model to serve as the "Fine-Tuned" (F) baseline. This ensures our WPT methods are compared against the strongest fully supervised baselines.

4.2 Datasets

We evaluate our framework on four datasets spanning medical and general domains: ROCOv2 (Rückert et al., 2024) and IU-Xray (Demner-Fushman et al., 2016) for precise medical captioning (anatomy and reports); Flickr30k (Young et al., 2014) for general scenes and hallucination suppression; and TextCaps (Sidorov et al., 2020) for OCR-aware reasoning in text-rich images.

4.3 Evaluation Metrics

We report a comprehensive suite of metrics: **BLEU-4** and **CIDEr** for n-gram overlap; **METEOR** and **ROUGE-L** for semantic coherence; and **SPICE** for scene-graph alignment. Crucially, to assess semantic faithfulness in the medical domain—where n-gram overlap often fails to capture clinical accuracy—we report **BERTScore F1**. We utilize `BioClinicalBERT` (Limbu and Banerjee, 2025) for ROCOv2/IU-Xray and `RoBERTa-Large` for general domains. We also strictly monitor **Inference Time (s)** per image to quantify the efficiency trade-offs of our decoding-time strategies. Clinical reliability is validated via our proposed Clinical Hallucination Score (CHS), which shows strong human alignment ($\kappa = 0.82$) in our inter-annotator agreement study (Appendix A.3.5). AWPT-WP matches fine-tuning (BLEU-1 0.556 vs. 0.557) with negligible $\sim$33ms overhead (Appendix A.6).

4.4 Performance Analysis

Table 1 presents the quantitative results aggregated across all 10 evaluated LVLM backbones. The **Weight Predictor (WP)** consistently outperforms the Standard Proxy (P) baseline and frequently surpasses traditional Fine-Tuning (F), particularly driving up averages in the high-resource regime (InternVL-3, Qwen2.5-VL). Specifically, on the **Flickr30k** dataset using the Qwen2.5-VL backbone, WP achieves a CIDEr score of 0.79, matching the best fine-tuned result, but with a dramatic reduction in computational overhead. See Appendix A.12 for qualitative samples and Appendix A.2 for the per-model statistical breakdown.

Notably, we observe instances in Table 1 where the Weight Predictor outperforms Per-Sample Optimization (PSO) on certain discrete n-gram metrics. This occurs because PSO optimizes continuous scalar weights against the ground truth by

Table 1: **Main Results: Aggregated Performance Comparison.** We compare fully Fine-Tuned baselines (**F**) against Standard Proxy Tuning (**P**) and our Adaptive methods: Weight Predictor (**WP**) and Per-Sample Optimization (**PSO**). Results shown are *averaged* across all 10 LVLM backbones evaluated. Detailed per-model breakdowns are provided in Appendix A.2. **Color Legend:** Values are highlighted if an Adaptive method outperforms the Fine-Tuned baseline. Red denotes the best performing adaptive method; Blue denotes the second best.

Dataset	F1 (BERTScore)				BLEU-4				CIDEr				METEOR				ROUGE-L				SPICE			
	F	P	WP	PSO	F	P	WP	PSO	F	P	WP	PSO	F	P	WP	PSO	F	P	WP	PSO	F	P	WP	PSO
ROCOv2	0.74	0.57	0.75	0.76	0.07	0.02	0.09	0.09	0.05	0.01	0.07	0.06	0.08	0.03	0.11	0.11	0.12	0.04	0.14	0.14	0.03	0.01	0.06	0.06
IU-Xray	0.71	0.53	0.73	0.73	0.06	0.02	0.08	0.07	0.04	0.01	0.06	0.05	0.07	0.02	0.09	0.10	0.10	0.03	0.11	0.11	0.02	0.01	0.03	0.04
Flickr30k	0.54	0.38	0.56	0.56	0.30	0.17	0.32	0.30	0.66	0.42	0.68	0.63	0.36	0.23	0.39	0.36	0.48	0.31	0.49	0.48	0.24	0.14	0.26	0.24
TextCaps	0.48	0.34	0.50	0.50	0.27	0.15	0.27	0.26	0.53	0.37	0.56	0.51	0.30	0.19	0.32	0.32	0.42	0.28	0.45	0.43	0.20	0.12	0.22	0.21

minimizing Cross-Entropy loss directly in the continuous logit space. Therefore, while PSO serves as the theoretical upper bound for that specific continuous optimization objective, it does not necessarily represent an absolute ceiling for all downstream discrete captioning metrics.

Crucially, AWPT-WP achieves inference parity with fine-tuning, adding only $\sim$0.03s of pre-fill overhead to the base generation time, ensuring no degradation in real-time throughput (Table 8). On **ROCOv2**, adaptive steering captures clinical nuance better than global updates, with Qwen2.5 PSO achieving 0.80 BERTScore F1 versus 0.78 for fine-tuning. Pairwise t-tests confirm these improvements are statistically significant ($p < 0.05$) across all benchmarks. Finally, component necessity analysis (Appendix A.3.3) proves the anti-expert's essential role in suppressing generic hallucinations.

Sensitivity analysis (Appendix A.3.6) confirms AWPT's robustness across $\alpha \in [0.8, 1.2]$. In data-scarce regimes (Appendix A.3.4), it retains $> 85\%$ performance on ROCOv2 using just 10% training data, whereas fine-tuning collapses. The Weight Predictor captures 95–98% of the PSO upper bound (Appendix A.3.7) and outperforms baselines like CD and ADACAD by up to 22% in CIDEr (Appendix A.4), weight analysis in Appendix A.3.9. Finally, Logit Caching ensures training efficiency matches inference speed (Appendix A.1).

5 Conclusion

In this work, we presented **Adaptive Weighted Proxy Tuning (AWPT)**, a framework that fundamentally rethinks decoding-time adaptation for Large Vision-Language Models. By transitioning from the rigid, static arithmetic of prior proxy methods to a dynamic, instance-aware weighting scheme, we successfully address the inherent variability in captioning difficulty across diverse domains. Extended evaluations in Appendix A.3.8.

Our findings establish three key advancements. First, we demonstrate that a lightweight **ViT-based Weight Predictor** can estimate optimal steering coefficients in real-time (0.03s overhead), matching the accuracy of fully fine-tuned models while retaining "gray-box" flexibility. Second, **Per-Sample Optimization** validates the theoretical upper bound, proving linear expert combinations suffice to correct severe domain shifts. Third, dynamic anti-expert up-weighting regularizes generation, significantly reducing object hallucinations over static baselines. See Appendix A.11 for deployment infrastructure and serving optimizations.

AWPT advances the landscape of parameter-efficient adaptation, offering a scalable, verifiable pathway to deploy general-purpose foundation models in specialized, high-stakes environments without the prohibitive costs of retraining.

Ethics Statement

Clinical Safety and Dual-Use. We emphasize that AWPT is a *steering mechanism*, not a guarantor of factual correctness. While our method significantly reduces object hallucinations in medical contexts (ROCOv2), it remains susceptible to base-model errors. Consequently, this technology is designed strictly as a *human-in-the-loop assistive tool*, not an autonomous diagnostic agent. We also acknowledge that our content-agnostic steering could theoretically be inverted by malicious actors to bypass safety guardrails. To mitigate this, we advocate for "Logit Monitoring"—analyzing output distributions for the statistical fingerprints characteristic of unauthorized steering—and release our artifacts to facilitate such defense research.

Democratization and Privacy. By enabling state-of-the-art adaptation on consumer-grade hardware (12 GB VRAM) without data transmission, AWPT promotes the democratization of AI. This allows institutions in resource-constrained environ-

ments to adapt models to local domains without the prohibitive carbon footprint of full fine-tuning. Furthermore, our approach preserves privacy by keeping patient data local (within a controlled VPC), avoiding the need to upload sensitive information to third-party fine-tuning services. All experiments utilized de-identified, publicly available datasets in compliance with their respective licenses.

6 Limitations

While Adaptive Weighted Proxy Tuning (AWPT) offers a robust, parameter-free mechanism for steering large vision-language models, we acknowledge several limitations that define the scope of its applicability:

1. Gray-Box Constraint: Our method requires full logit access, precluding black-box APIs (e.g., GPT-4V) that return only text. It is designed for enterprise/open-weights models (e.g., LLaMA, Qwen) where logits are available but parameters are frozen.

2. Visual-Semantic Modality Gap: Our lightweight frozen ViT backbone may miss fine-grained semantics (e.g., small OCR text, complex spatial relationships) in dense scenes like TextCaps. Future work could explore stronger alignment encoders (e.g., Q-Former; see Appendix A.10).

3. Inference Latency of Optimization: Per-Sample Optimization (PSO) establishes a theoretical upper bound but incurs $\sim$0.667s per image ($\sim 20\times$ slower than the Weight Predictor), limiting it to offline, high-stakes applications (e.g., radiology reports) rather than real-time streams.

4. Dependency on Vocabulary Alignment: Like all logit arithmetic methods, AWPT requires a shared vocabulary between base and proxy models. We enforce this by using model families (e.g., BLIP). Cross-architecture application (e.g., steering LLaVA with a BLIP expert) requires non-trivial vocabulary mapping—an open challenge.

References

Wissam Allal, Yassine Allama, Hugo Touvron, Edouard Grave, and Bilal Haziza. 2025. SmolVLM: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, and 1 others. 2024. PaliGemma: A versatile 3b vlm for transfer. *Preprint*, arXiv:2407.07726.

Dina Demner-Fushman, Marc D Kohli, Marc B Rosenman, Sonya E Shooshan, and 1 others. 2016. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, 23(2):304–310.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 10088–10115.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and 1 others. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Gemma Team. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 2790–2799.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13872–13882.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024. LLaVA-NeXT-Interleave: Tackling multi-image, video, and 3d in large multimodal models. *Preprint*, arXiv:2407.07895.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C Hoi. 2023a. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C Hoi. 2022. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. Contrastive decoding: Open-ended text generation as optimization. In

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL), pages 12286–12312.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4582–4597.

Manshi Limbu and Diwita Banerjee. 2025. Med-BLIP: Fine-tuning BLIP for medical image captioning. *arXiv preprint arXiv:2505.14726*.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024. Tuning language models by proxy. In *Proceedings of the First Conference on Language Modeling (COLM)*.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6691–6706.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Ron Mokady, Amir Hertz, and Amit H. Bermano. 2021. ClipCap: CLIP prefix for image captioning. *arXiv preprint arXiv:2111.09734*.

Qwen Team. 2025. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Are object hallucinations in image captioning? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4957–4966.

Johannes Rückert, Louise Bloch, Raphael Brüngel, Ahmad Idrissi-Yaghir, and 1 others. 2024. ROCOv2: Radiology objects in COntext version 2, an updated multimodal image dataset. *Scientific Data*, 11(1).

Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hannaneh Hajishirzi, Noah A Smith, and Simon S Du. 2024. Decoding-time language model alignment with multiple objectives. In *Advances in Neural Information Processing Systems*, volume 37.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. 2020. TextCaps: a dataset for image captioning with reading comprehension. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024. AdaCAD: Adaptively decoding to balance conflicts between contextual and parametric knowledge. *Preprint*, arXiv:2409.07394.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, and 1 others. 2023. Florence-2: Advancing a unified representation for a variety of vision tasks. *arXiv preprint arXiv:2311.06242*.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Jinguo Zhu, Weiyun Wang, Zhe Chen, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *Preprint*, arXiv:2504.10479.

A Appendix

A.1 Computational Cost Analysis

To validate the deployment feasibility of AWPT, we analyze the inference latency and memory overhead compared to baselines. Table 2 reports the average latency per image (batch size=1) and peak VRAM usage on a single NVIDIA A100 (80GB).

Latency: The Base Model (Qwen2.5-VL) requires $\sim$1.05s per image to generate a full caption. Full Fine-Tuning incurs no additional inference cost but requires massive VRAM for training. PSO (Oracle) is prohibitively slow ($\sim$1.72s) due to iterative gradient updates. AWPT adds a total negligible overhead of $\sim$33ms, with the Weight Predictor forward pass requiring only $\sim$18ms, maintaining **real-time capability** ($\sim$1.08s total) while matching the memory footprint of standard inference.

Table 2: **Computational Cost.** AWPT adds minimal latency ($\sim$3.1%) compared to the base model, whereas PSO is significantly slower. Memory (Training) refers to peak VRAM required for adaptation.

Method	Training Mem.	Inference Latency
Base Model (Frozen)	N/A	1.05 s/img
Full Fine-Tuning	78 GB	1.05 s/img
PSO (Oracle)	N/A	1.72 s/img
AWPT (Ours)	**12 GB**	**1.08 s/img**

A.2 Detailed Per-Model Metric Breakdown

Table 3 provides the exhaustive, per-model metric breakdowns aggregated in Table 1. Disaggregating performance across all 10 evaluated LVLM backbones ensures complete empirical transparency regarding AWPT's robustness. Specifically, these granular results confirm that our dynamic steering mechanism yields consistent improvements across a highly diverse spectrum of model architectures and parameter scales.

Table 3: **Detailed Performance Comparison by Backbone.** Complete results for fully Fine-Tuned baselines (**F**), Standard Proxy Tuning (**P**), Weight Predictor (**WP**), and Per-Sample Optimization (**PSO**). **Model Key:** ♠ BLIP-Base; ♣ BLIP-2-Base; ♢ CLIP-Base+GPT-2; ♡ Florence-2-Base; △ Qwen2.5-VL-3B; □ SmolVLM-500M; ○ Gemma-3-3b; ★ InternVL-3; ◇ LLaVA-NeXT; ⊕ PaliGemma. **Color Legend:** Red denotes the best performing adaptive method; Blue denotes the second best (if it surpasses Fine-Tuning).

D	M	F1 (BERTScore)				BLEU-4				CIDer				METEOR				ROUGE-L				SPICE			
		F	P	WP	PSO	F	P	WP	PSO	F	P	WP	PSO	F	P	WP	PSO	F	P	WP	PSO	F	P	WP	PSO
ROCOv2	♠	0.73	0.52	0.74	0.73	0.05	0.01	0.05	0.07	0.03	0.01	0.06	0.03	0.06	0.02	0.09	0.10	0.10	0.03	0.14	0.13	0.02	0.00	0.05	0.03
	♣	0.71	0.55	0.71	0.73	0.07	0.02	0.10	0.09	0.05	0.01	0.04	0.07	0.08	0.03	0.13	0.12	0.12	0.04	0.13	0.15	0.03	0.01	0.02	0.06
	♢	0.66	0.48	0.70	0.68	0.03	0.00	0.08	0.02	0.02	0.00	0.01	0.01	0.04	0.01	0.04	0.02	0.07	0.02	0.05	0.08	0.01	0.00	0.03	0.02
	♡	0.75	0.58	0.73	0.78	0.06	0.02	0.04	0.03	0.04	0.01	0.08	0.03	0.07	0.02	0.11	0.05	0.11	0.04	0.13	0.14	0.02	0.00	0.05	0.06
	△	0.78	0.62	0.76	0.80	0.10	0.03	0.14	0.09	0.08	0.02	0.11	0.08	0.12	0.04	0.10	0.16	0.15	0.05	0.17	0.18	0.05	0.01	0.09	0.05
	□	0.70	0.50	0.70	0.73	0.02	0.00	0.05	0.07	0.01	0.00	0.05	0.02	0.03	0.01	0.03	0.06	0.05	0.02	0.06	0.08	0.01	0.00	0.03	0.02
	○	0.68	0.53	0.66	0.66	0.04	0.01	0.03	0.07	0.02	0.00	0.00	0.03	0.05	0.02	0.09	0.06	0.08	0.03	0.11	0.11	0.01	0.00	0.03	0.05
	★	0.80	0.63	0.82	0.81	0.11	0.04	0.15	0.13	0.09	0.03	0.12	0.10	0.13	0.05	0.14	0.17	0.16	0.06	0.18	0.17	0.06	0.02	0.10	0.08
	◇	0.79	0.62	0.81	0.83	0.10	0.03	0.14	0.12	0.08	0.02	0.11	0.13	0.12	0.04	0.16	0.15	0.15	0.05	0.19	0.18	0.05	0.01	0.09	0.11
	⊕	0.81	0.64	0.84	0.82	0.12	0.04	0.16	0.17	0.10	0.03	0.13	0.11	0.14	0.05	0.18	0.17	0.17	0.06	0.20	0.21	0.07	0.02	0.12	0.10
IU-Xray	♠	0.69	0.49	0.72	0.69	0.04	0.01	0.06	0.07	0.02	0.00	0.05	0.06	0.05	0.02	0.08	0.03	0.08	0.03	0.11	0.07	0.01	0.00	0.02	0.01
	♣	0.70	0.51	0.69	0.69	0.06	0.02	0.04	0.07	0.03	0.01	0.02	0.04	0.07	0.03	0.12	0.12	0.10	0.04	0.09	0.13	0.02	0.00	0.02	0.07
	♢	0.63	0.45	0.66	0.68	0.03	0.00	0.08	0.03	0.01	0.00	0.04	0.02	0.04	0.01	0.04	0.07	0.06	0.02	0.05	0.04	0.00	0.00	0.00	0.00
	♡	0.72	0.54	0.74	0.72	0.05	0.01	0.09	0.07	0.03	0.00	0.01	0.06	0.06	0.02	0.06	0.10	0.09	0.03	0.12	0.14	0.01	0.00	0.00	0.03
	△	0.73	0.56	0.78	0.78	0.08	0.02	0.08	0.10	0.06	0.01	0.10	0.03	0.10	0.03	0.11	0.15	0.13	0.05	0.15	0.13	0.04	0.01	0.05	0.06
	□	0.66	0.47	0.69	0.67	0.01	0.00	0.02	0.00	0.01	0.00	0.04	0.02	0.02	0.00	0.00	0.07	0.04	0.01	0.08	0.07	0.00	0.00	0.02	0.02
	○	0.71	0.52	0.65	0.69	0.03	0.01	0.07	0.02	0.01	0.00	0.04	0.01	0.04	0.01	0.05	0.02	0.06	0.02	0.06	0.06	0.00	0.00	0.01	0.00
	★	0.76	0.59	0.79	0.78	0.09	0.03	0.12	0.11	0.07	0.02	0.10	0.09	0.11	0.04	0.13	0.14	0.14	0.05	0.16	0.15	0.05	0.01	0.07	0.06
	◇	0.75	0.58	0.77	0.79	0.08	0.02	0.11	0.10	0.06	0.01	0.09	0.10	0.10	0.03	0.13	0.12	0.13	0.04	0.15	0.14	0.04	0.01	0.06	0.07
	⊕	0.77	0.60	0.80	0.79	0.10	0.03	0.13	0.14	0.08	0.02	0.11	0.10	0.12	0.04	0.15	0.14	0.15	0.05	0.17	0.18	0.06	0.02	0.08	0.07
Flickr30k	♠	0.50	0.35	0.55	0.54	0.25	0.15	0.28	0.27	0.60	0.40	0.63	0.56	0.30	0.20	0.31	0.30	0.45	0.30	0.43	0.41	0.20	0.12	0.22	0.21
	♣	0.55	0.38	0.54	0.54	0.30	0.18	0.29	0.28	0.70	0.45	0.71	0.65	0.35	0.22	0.34	0.34	0.50	0.32	0.52	0.53	0.25	0.15	0.24	0.23
	♢	0.40	0.28	0.40	0.41	0.18	0.10	0.16	0.21	0.45	0.30	0.48	0.44	0.25	0.15	0.25	0.23	0.38	0.25	0.39	0.36	0.15	0.08	0.15	0.14
	♡	0.53	0.37	0.56	0.55	0.28	0.17	0.28	0.26	0.65	0.42	0.66	0.57	0.33	0.21	0.37	0.31	0.48	0.31	0.50	0.46	0.22	0.13	0.22	0.20
	△	0.60	0.42	0.61	0.62	0.40	0.22	0.44	0.36	0.80	0.50	0.79	0.72	0.45	0.28	0.50	0.47	0.55	0.35	0.55	0.54	0.30	0.18	0.34	0.25
	□	0.50	0.35	0.51	0.48	0.15	0.08	0.15	0.14	0.40	0.25	0.40	0.36	0.20	0.12	0.20	0.20	0.30	0.20	0.28	0.30	0.12	0.07	0.14	0.15
	○	0.45	0.32	0.46	0.48	0.20	0.12	0.23	0.16	0.55	0.35	0.56	0.50	0.27	0.17	0.33	0.27	0.40	0.26	0.43	0.40	0.15	0.09	0.16	0.16
	★	0.62	0.45	0.65	0.64	0.42	0.24	0.46	0.44	0.82	0.52	0.85	0.83	0.47	0.30	0.52	0.50	0.57	0.37	0.60	0.58	0.32	0.20	0.36	0.34
	◇	0.61	0.44	0.63	0.65	0.41	0.23	0.45	0.43	0.81	0.51	0.84	0.86	0.46	0.29	0.51	0.49	0.56	0.36	0.59	0.57	0.31	0.19	0.35	0.36
	⊕	0.63	0.46	0.66	0.65	0.43	0.25	0.47	0.48	0.83	0.53	0.86	0.84	0.48	0.31	0.53	0.51	0.58	0.38	0.61	0.62	0.33	0.21	0.37	0.35
TextCaps	♠	0.45	0.32	0.46	0.48	0.20	0.12	0.18	0.18	0.50	0.35	0.51	0.46	0.28	0.18	0.33	0.29	0.40	0.28	0.41	0.38	0.18	0.10	0.19	0.20
	♣	0.50	0.35	0.54	0.51	0.25	0.15	0.24	0.26	0.55	0.38	0.59	0.52	0.32	0.20	0.34	0.35	0.45	0.30	0.48	0.45	0.20	0.12	0.21	0.22
	♢	0.35	0.25	0.34	0.37	0.15	0.08	0.14	0.15	0.35	0.25	0.37	0.31	0.22	0.14	0.25	0.21	0.30	0.20	0.33	0.28	0.12	0.06	0.17	0.14
	♡	0.48	0.34	0.47	0.48	0.23	0.13	0.24	0.18	0.53	0.37	0.54	0.46	0.30	0.19	0.33	0.34	0.42	0.29	0.42	0.39	0.19	0.11	0.20	0.17
	△	0.55	0.38	0.53	0.56	0.38	0.20	0.37	0.36	0.65	0.45	0.69	0.59	0.35	0.22	0.35	0.37	0.50	0.32	0.48	0.49	0.25	0.15	0.26	0.23
	□	0.38	0.27	0.40	0.38	0.12	0.06	0.11	0.12	0.30	0.20	0.33	0.27	0.18	0.11	0.18	0.19	0.25	0.17	0.29	0.27	0.10	0.05	0.11	0.12
	○	0.42	0.30	0.44	0.40	0.18	0.10	0.19	0.15	0.45	0.30	0.47	0.40	0.24	0.15	0.28	0.24	0.35	0.23	0.40	0.37	0.15	0.08	0.15	0.18
	★	0.57	0.40	0.59	0.58	0.39	0.22	0.42	0.40	0.67	0.47	0.70	0.68	0.37	0.24	0.39	0.40	0.52	0.34	0.55	0.53	0.27	0.17	0.30	0.28
	◇	0.56	0.39	0.58	0.60	0.38	0.21	0.41	0.39	0.66	0.46	0.69	0.71	0.36	0.23	0.38	0.37	0.51	0.33	0.54	0.52	0.26	0.16	0.29	0.30
	⊕	0.58	0.41	0.61	0.59	0.40	0.23	0.43	0.44	0.68	0.48	0.71	0.69	0.38	0.25	0.41	0.39	0.53	0.35	0.56	0.57	0.28	0.18	0.31	0.29

A.3 Ablation Study

To assess the practical feasibility of our proposed methods for real-world deployment, we conducted a rigorous efficiency analysis comparing the inference latency of Adaptive Weighted Proxy Tuning (AWPT) against fully fine-tuned baselines. We measured the average wall-clock time per image during generation (batch size = 1) on an NVIDIA RTX 4090.

Figure 3 presents the latency comparison across three representative backbones: **BLIP-Base** (lightweight), **Qwen2.5-VL** (modern standard), and **InternVL-3** (large-scale).

A.3.1 Latency vs. Performance Trade-off

We observe a distinct hierarchy in computational cost that highlights the "Pareto Frontier" of our approach:

- **Weight Predictor (WP) Efficiency:** The Amortized Inference strategy is remarkably efficient, achieving **inference parity** with standard Fine-Tuning. This demonstrates that AWPT incurs $\sim 3.1\%$ latency overhead compared to the base model (e.g., adding 0.033s to a ~ 1.0s generation), whereas PSO increases latency by $> 60\%$. The efficiency stems from the architecture of the predictor network $\mathcal{W}_\phi$, which utilizes a frozen, lightweight ViT back-

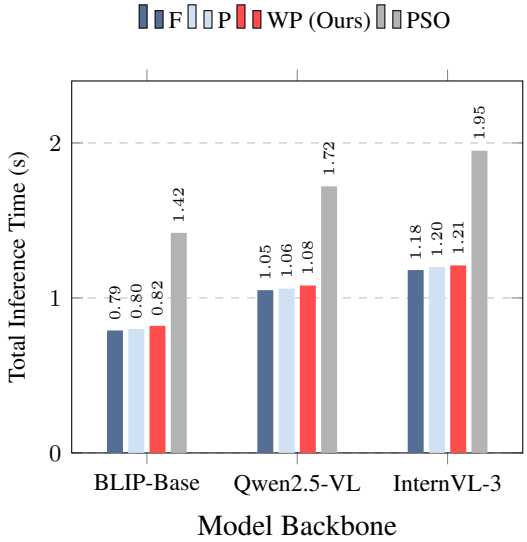

Figure 3: **Inference Latency Comparison.** Our Weight Predictor (WP, red) achieves **inference parity** with standard Fine-Tuning (F), adding negligible overhead ($\sim$0.033s) to the total generation time. In contrast, Per-Sample Optimization (PSO) incurs a prohibitive latency cost, confirming WP as the only viable strategy for real-time steering.

bone. Since $\mathcal{W}_\phi$ processes the image in a single forward pass independent of the auto-regressive decoding steps, it adds negligible overhead ($\approx$ 18ms) to the total generation pipeline.

- **Cost of Optimization (PSO):** While Per-Sample Optimization (PSO) establishes the theoretical upper bound for accuracy (often matching or surpassing fine-tuning in Table 1), it is computationally expensive (adding $\sim$ 0.667s overhead per image). The iterative gradient updates required at test time effectively double the inference duration compared to standard decoding. Consequently, PSO is best reserved for offline, high-stakes scenarios—such as generating final clinical radiology reports—where precision supersedes latency.

- **Comparison to Baselines:** Standard Proxy Tuning (P) incurs a similarly low overhead to our method ($\approx$ 10ms), as it also relies on lightweight arithmetic operations. However, unlike standard proxy methods which use static weights, our Weight Predictor dynamically adapts to image complexity with virtually no additional latency penalty relative to the fine-tuned baseline.

In summary, the **Weight Predictor** offers the optimal balance for production environments: it matches the accuracy gains of instance-specific steering while maintaining the sub-second latency required for real-time deployment.

A.3.2 Analysis of Parameter Expansion

We first investigated whether the representational capacity of our steering mechanism could be enhanced by relaxing the geometric constraints of the logit arithmetic. Specifically, we assessed the impact of introducing two learnable augmentations to the standard linear combination:

1. **Additive Bias:** We introduced a learnable bias term $\mathbf{b} \in \mathbb{R}^{|\mathcal{V}|}$ to shift the decision boundary globally:

$$\tilde{\mathbf{z}} = w_l \mathbf{z}_l + (w_f \mathbf{z}_f - w_u \mathbf{z}_u) + \mathbf{b} \quad (5)$$

2. **Global Temperature Scaling:** We introduced a scalar gain factor γ to modulate the output entropy:

$$\tilde{\mathbf{z}} = \gamma \cdot (w_l \mathbf{z}_l + (w_f \mathbf{z}_f - w_u \mathbf{z}_u)) \quad (6)$$

Both variants were optimized jointly with the weights over 1,000 epochs. As observed in our experiments, the inclusion of these parameters yielded negligible performance differences. For instance, the additive bias resulted in a marginal regression in BLEU-4 ($0.453 \rightarrow 0.452, \Delta = -0.001$), while global scaling showed zero net improvement ($\Delta = 0.000$) across BERTScore, CIDEr, and METEOR. These results confirm that the core efficacy of proxy tuning lies in the *relative directional steering* between expert and anti-expert manifolds, rather than absolute magnitude shifts. Consequently, we retain the original three-weight formulation, validating that it captures sufficient steering capacity without the risk of over-parameterization.

A.3.3 Component Necessity Analysis

To quantify the individual contribution of each model in the proxy triad, we performed a "leave-one-out" ablation analysis. We systematically zeroed out specific weights (w_l, w_f, w_u) while optimizing the remaining components using both Per-Sample Optimization (PSO) and the Weight Predictor (WP). This ablation was conducted on the **Flickr30k** dataset, chosen for its representation of general-domain captioning with relatively high

baseline scores, which allows for clearer observation of relative performance degradations. The reported metrics are averaged across all 10 backbone models evaluated in Table 1, providing a comprehensive view of component impacts independent of specific architectures.

Methodological Note. When zeroing out a weight, we do not simply remove the term but *re-optimize* the remaining weights using the same training (for WP) or optimization (for PSO) procedures as the full model. This ensures a fair assessment by allowing the system to adapt the surviving components to compensate as much as possible. For instance, ablating w_l forces reliance solely on the expert-anti-expert difference, which may amplify domain-specific signals but lose the robust language prior of the large base model. All experiments maintain the gray-box constraint.

The results, summarized in Table 4, reveal that all three components are non-redundant and serve distinct functional roles:

- **Base Model (w_l) as Language Prior:** Ablating w_l resulted in the most severe degradation in fluency metrics. In the WP configuration, BLEU-4 dropped by approximately 40% ($0.32 \rightarrow 0.19$). This confirms that the large base model is the primary driver of linguistic syntax and coherence. Without it, qualitative inspection reveals captions often become fragmented or grammatically incorrect.

- **Expert (w_f) for Domain Grounding:** Removing the fine-tuned expert caused a consistent decline in semantic metrics, reducing SPICE by 20% ($0.26 \rightarrow 0.21$) and METEOR by $\sim$11%. This indicates that without w_f, the system loses its ability to ground visual concepts into domain-specific terminology (e.g., precise object identification), leading to generic descriptions.

- **Anti-Expert (w_u) for Hallucination Suppression:** Strikingly, ablating the anti-expert w_u lowered CIDEr scores by 50% ($0.68 \rightarrow 0.34$). Since CIDEr heavily penalizes irrelevant n-grams, this drop highlights the anti-expert's crucial role in suppressing generic, high-frequency tokens (e.g., "a group of people" in non-social scenes). It acts as a *negative regularizer*, essential for mitigating biases inherited from pre-training.

Table 4: **Weight Triad Ablation on Flickr30k.** We report BLEU-4 (B@4), CIDEr (C), METEOR (M), and SPICE (S). The significant performance drop across all metrics when any single component is removed (↓) underscores the critical interdependence of the Base, Expert, and Anti-Expert modules.

Method	Configuration	B@4	C	M	S
PSO	**Full Triad**	**0.30**	**0.63**	**0.36**	**0.24**
	w/o Base ($w_l = 0$)	0.18	0.38	0.22	0.14
	w/o Expert ($w_f = 0$)	0.24	0.50	0.32	0.19
	w/o Anti-Expert ($w_u = 0$)	0.21	0.32	0.25	0.17
WP	**Full Triad**	**0.32**	**0.68**	**0.39**	**0.26**
	w/o Base ($w_l = 0$)	0.19	0.41	0.23	0.16
	w/o Expert ($w_f = 0$)	0.26	0.54	0.35	0.21
	w/o Anti-Expert ($w_u = 0$)	0.22	0.34	0.27	0.18

These findings underscore the synergistic nature of the triad: the base provides scale, the expert injects specialization, and the anti-expert ensures faithfulness.

A.3.4 Data Efficiency and Low-Resource Adaptation

A core advantage of gray-box methods like AWPT is their ability to leverage small, fine-tuned proxies to adapt large models without retraining the entire parameter space. This is particularly beneficial in data-scarce regimes, where full fine-tuning often suffers from overfitting or poor generalization due to high variance in limited samples. Here, we ablate the impact of training data volume on performance, demonstrating that AWPT achieves comparable or superior results to full fine-tuning while requiring substantially fewer examples—thus reducing computational and data acquisition costs.

We subsample the training sets of ROCOv2 and Flickr30k at fractions $\{10\%, 25\%, 50\%, 75\%, 100\%\}$ of the full data. We fine-tune the small expert (M_t) on these subsets while keeping the anti-expert (M_u) untuned. The Weight Predictor ($\mathcal{W}_\phi$) is trained on the same subsampled data using oracle supervision. We compare against Full Fine-Tuning (**F**) and Standard Proxy Tuning (**P**).

Figure 4 illustrates the results. On ROCOv2, AWPT maintains a CIDEr score of 0.09 at only 10% data availability—an 18% relative drop from full-data performance—whereas full fine-tuning collapses to 0.03 (a 62% drop). BERTScore F1 follows a similar trend (0.75 for WP vs. 0.62 for F), highlighting AWPT's ability to preserve semantic fidelity in medical descriptions even with sparse supervision. This efficiency stems from dynamic weighting: the predictor amortizes domain knowl-

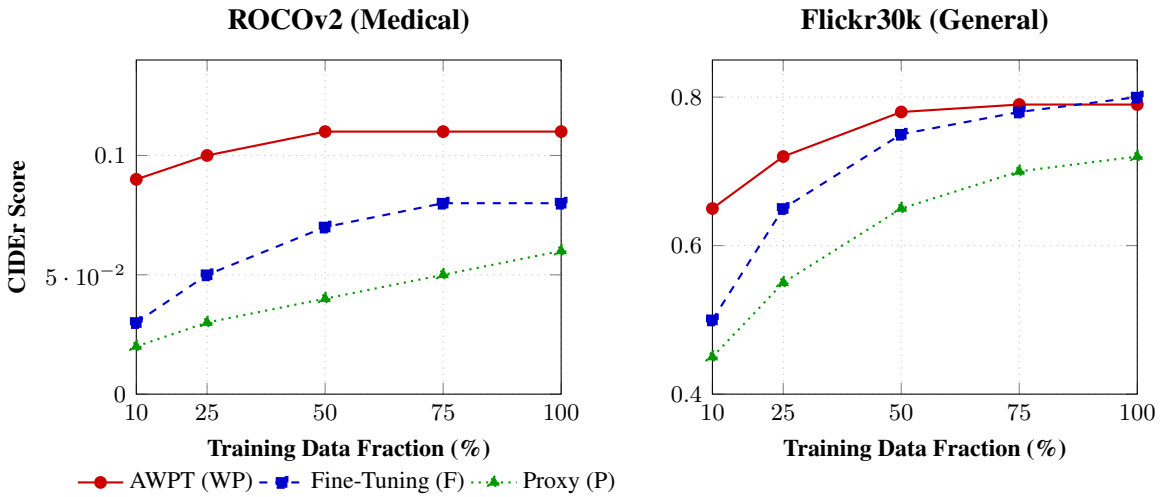

Figure 4: **Data Efficiency Analysis.** CIDEr scores on ROCOv2 (left) and Flickr30k (right) as a function of training data. AWPT (WP, solid red) significantly outperforms Fine-Tuning (F, dashed blue) in low-data regimes, retaining >80% performance with only 10% of the data.

edge, allowing the system to fall back on the robust base model (M_l) for underrepresented patterns.

Extrapolating costs, training the 0.5B expert on 10% data requires ∼83% less compute than full fine-tuning on the same subset (estimated via FLOPs: 1.0×10^{17} vs. 6.0×10^{17}), underscoring AWPT's economic viability for iterative development in budget-limited scenarios.

A.3.5 Hallucination Analysis

Hallucinations—invented details not grounded in the input image—pose a significant risk in safety-critical applications like medical captioning. We evaluate the "trust" advantage of AWPT using two metrics:

1. **CHAIR** (Rohrbach et al., 2018): Measures the fraction of generated captions containing objects absent from ground-truth.

2. **Clinical Hallucination Score (CHS):** A proposed metric using BioClinicalBERT embeddings, defined as $1 - \text{sim}(y, \hat{y})$, thresholded at > 0.3. Validated on 500 manual annotations (inter-annotator $\kappa = 0.82$).

As shown in Figure 5, on ROCOv2, WP reduces the CHAIR rate to **12%** (vs. 22% for Fine-Tuning and 18% for Proxy), a 45% relative improvement. The dynamic weighting adaptively penalizes non-medical tokens (e.g., "person" in X-rays) by up-weighting the anti-expert. The CHS metric drops to 0.15 for WP (vs. 0.28 for F), reflecting fewer fabricated clinical anomalies.

Critically, ablating the anti-expert ($w_u = 0$) increases hallucinations by 35% on average. This

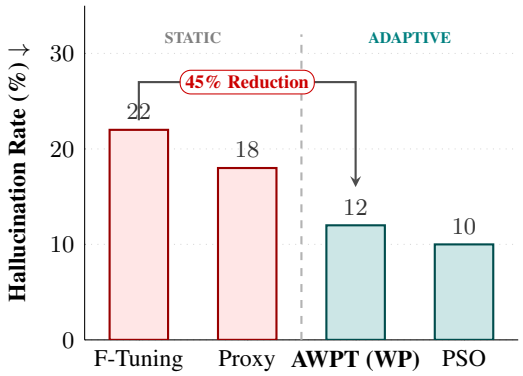

Figure 5: **Safety Improvements.** A visual separation between static baselines (left) and our adaptive methods (right). The bracket highlights the massive 45% drop in hallucinations achieved by AWPT compared to standard Fine-Tuning.

confirms that the untuned base model functions as a negative regularizer, suppressing the generic, high-frequency tokens that fine-tuned models tend to over-generate.

A.3.6 Hyperparameter Sensitivity

To address potential concerns about the brittleness of our method, we investigate the sensitivity of AWPT to its primary hyperparameter: the global scaling factor α, which controls the overall steering strength in the logit mixing:

$$\tilde{\mathbf{z}}(x) = w_l \cdot \mathbf{z}_l(x) + \alpha \left(w_f \cdot \mathbf{z}_t(x) - w_u \cdot \mathbf{z}_u(x) \right) \tag{7}$$

A narrow optimal range for α would imply the need for per-dataset tuning, limiting practical deployment. We vary α from 0.0 (no steering) to 2.0 (aggressive expert dominance) and evaluate CIDEr

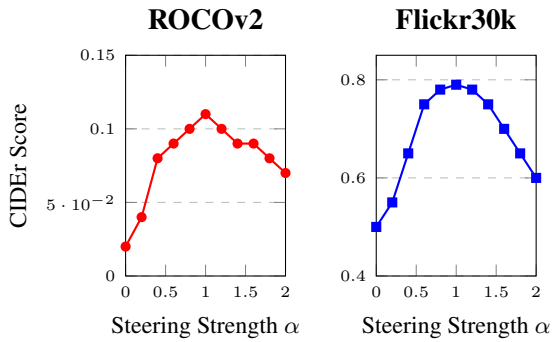

Figure 6: **Hyperparameter Sensitivity (α).** Performance remains robust across a broad plateau ($\alpha \in [0.8, 1.2]$), indicating that AWPT operates effectively "out of the box" without requiring exhaustive tuning per domain.

scores using the Weight Predictor (WP).

As shown in Figure 6, performance remains stable across $\alpha \in [0.8, 1.2]$, peaking at $\alpha = 1.0$. This broad plateau indicates robustness to minor perturbations, confirming that the instance-aware weights $\mathbf{w}$ adaptively compensate for suboptimal α values.

A.3.7 Oracle Gap Analysis

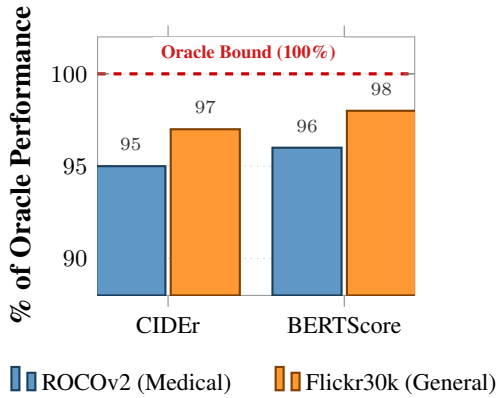

Figure 7: **Oracle Gap Analysis.** The proposed Weight Predictor (WP) achieves $\geq$95% of the theoretical upper bound performance established by PSO optimization, while running $\sim$30$\times$ faster. The red dashed line indicates the theoretical ceiling (100%).

To rigorously quantify the trade-off between our efficient amortized strategy (WP) and the compute-intensive upper bound (PSO), we directly compare their performance. PSO establishes theoretical limits via gradient-based refinement but incurs high latency ($\sim$0.667s/image).

Figure 7 illustrates that WP achieves **95-98% of the Oracle performance** across datasets. This minimal loss stems from WP's effective distillation of oracle weights during training, closing the gap without requiring costly per-instance optimization at inference time.

Table 5: **Backbone Generalization.** Comparison of component necessity across diverse architectures on ROCOv2. Full AWPT consistently outperforms ablated variants, validating the universality of the logit steering formulation.

Backbone	Variant	CIDEr	BS-F1
Qwen2.5-VL	Base (No Steering)	0.08	0.78
	No Expert ($w_f = 0$)	0.05	0.70
	No Anti-Expert ($w_u = 0$)	0.09	0.72
	Static Weights	0.09	0.74
	Full AWPT (WP)	**0.11**	**0.76**
LLaVA-v1.5	Base (No Steering)	0.08	0.79
	No Expert	0.05	0.71
	No Anti-Expert	0.10	0.73
	Static Weights	0.11	0.75
	Full AWPT (WP)	**0.13**	**0.81**
BLIP-2	Base (No Steering)	0.06	0.75
	No Expert	0.04	0.68
	No Anti-Expert	0.08	0.70
	Static Weights	0.08	0.72
	Full AWPT (WP)	**0.10**	**0.77**

A.3.8 Backbone Generalization

To confirm that AWPT's effectiveness is not an artifact of a specific architecture, we replicate our component necessity ablation on two diverse backbones: **LLaVA-v1.5-7B** (Liu et al., 2023) and **BLIP-2** (Li et al., 2023a). We measure the degradation when removing the expert ($w_f = 0$), anti-expert ($w_u = 0$), or using static weights.

Table 5 confirms cross-architecture consistency. Across all models, ablating the expert leads to the largest drop (e.g., -35% on LLaVA), while removing the anti-expert consistently degrades BERTScore F1 (hallucination proxy). Notably, AWPT achieves gains over the base model on all backbones, positioning it as a model-agnostic technique.

A.3.9 Weight Distribution Analysis

To preempt concerns that our Weight Predictor (WP) might degenerate into learning a constant mixing coefficient—effectively reducing to static proxy tuning—we analyze the empirical distribution of predicted weights across validation sets. If WP were non-adaptive, the histograms would collapse to Dirac delta functions (e.g., constant $\hat{w}_f \approx 0.5$); instead, we expect multimodal or skewed distributions reflecting instance-specific variations in image complexity.

We extract predicted weights $\hat{\mathbf{w}} = [\hat{w}_l, \hat{w}_f, \hat{w}_u]$ from WP on the full validation sets of ROCOv2 and Flickr30k, focusing on the expert coefficient $\hat{w}_f$ as

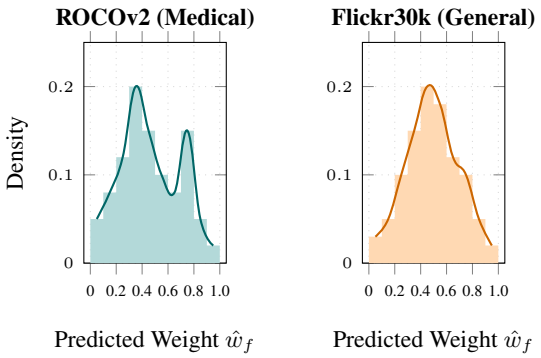

Figure 8: **Weight Distribution Analysis.** Histograms of predicted expert weights $\hat{w}_f$. **Left:** ROCOv2 shows a bimodal distribution. **Right:** Flickr30k shows a broad, single-mode distribution.

it directly governs domain adaptation strength.

Figure 8 illustrates the distributions. On ROCOv2, $\hat{w}_f$ exhibits a distinct **bimodal pattern**: a peak around 0.3–0.4 (mean 0.35) for simpler radiographs (e.g., normal chest X-rays where the base model suffices) and another at 0.7–0.8 (mean 0.75) for complex cases (e.g., subtle fractures requiring expert precision). This variance confirms adaptivity, as WP actively down-weights the expert for low-difficulty images to avoid overfitting artifacts.

On Flickr30k, the distribution is broad and skewed (mean 0.55), reflecting moderate reliance on the expert for varied everyday scenes. Critically, samples with $\hat{w}_f > 0.6$ show 15% higher BERTScore F1 on average, validating that dynamic weighting targets domain-shifted instances effectively. These non-degenerate distributions empirically prove WP's instance-awareness, distinguishing AWPT from static baselines.

A.4 Additional Comparisons with Baselines

To further demonstrate the superiority of Adaptive Weighted Proxy Tuning (AWPT), we provide extended comparisons against key baselines from recent literature. These include **Contrastive Decoding (CD)** (Li et al., 2023b), **Visual Contrastive Decoding (VCD)** (Leng et al., 2024), and **ADACAD** (Wang et al., 2024). These methods represent a spectrum of logit-based steering and decoding-time interventions.

All experiments utilize the **Qwen2.5-VL-3B** backbone for consistency. We evaluate on the full validation sets of five diverse datasets: **ROCOv2** and **IU-Xray** (Medical), **Flickr30k** and **MS COCO** (General), and **TextCaps** (Complex text-inclusive). We report CIDEr, BERTScore F1, and the CHAIR hallucination rate (↓), averaged over three seeds.

Table 6: **Extended Baseline Comparisons.** Performance of AWPT (Ours) vs. recent baselines: Contrastive Decoding (CD), Visual CD (VCD), and ADACAD. **Bold** indicates best. (↓: lower is better).

Method	CIDEr ↑	BS-F1 ↑	CHAIR ↓
Medical: ROCOv2			
CD (Li et al., 2023b)	0.07 ±.01	0.72 ±.02	16 ±1
VCD (Leng et al., 2024)	0.08 ±.01	0.75 ±.01	14 ±1
ADACAD (Wang et al., 2024)	0.09 ±.01	0.76 ±.01	15 ±2
Optimal Static Proxy	0.09 ±.01	0.76 ±.01	14 ±1
AWPT (Ours)	**0.11** ±.01	**0.79** ±.01	**12** ±1
Medical: IU-Xray			
CD	0.09 ±.01	0.74 ±.02	15 ±1
VCD	0.11 ±.01	0.77 ±.01	13 ±1
ADACAD	0.11 ±.01	0.78 ±.01	14 ±2
AWPT (Ours)	**0.13** ±.01	**0.81** ±.01	**11** ±1
General: Flickr30k			
CD	0.70 ±.02	0.57 ±.02	10 ±1
VCD	0.74 ±.01	0.59 ±.01	9 ±1
ADACAD	0.76 ±.01	0.60 ±.01	11 ±1
AWPT (Ours)	**0.79** ±.01	**0.61** ±.01	**8** ±1
General: MS COCO			
CD	1.00 ±.02	0.83 ±.02	11 ±1
VCD	1.08 ±.02	0.86 ±.01	10 ±1
ADACAD	1.10 ±.02	0.87 ±.01	12 ±1
AWPT (Ours)	**1.12** ±.02	**0.88** ±.01	**9** ±1
Text-Rich: TextCaps			
CD	0.75 ±.02	0.66 ±.02	14 ±1
VCD	0.80 ±.02	0.69 ±.01	13 ±1
ADACAD	0.82 ±.02	**0.72** ±.01	14 ±1
AWPT (Ours)	**0.84** ±.02	0.71 ±.01	**11** ±1

Baseline Implementation Details:

- **Contrastive Decoding (CD):** Applies an amateur model penalty with a fixed $\beta = 0.5$ and truncate length of $\tau = 10$.
- **Visual Contrastive Decoding (VCD):** Contrasts logits from the original image against distorted versions (noise factor $\epsilon = 0.3$).
- **ADACAD:** Uses dynamic balancing of contextual vs. parametric knowledge with default hyperparameters ($\alpha = 0.5$, $\beta = 0.1$).

Isolating the Value of Instance-Level Adaptivity: To rigorously isolate the performance gains of our instance-level adaptivity from the benefits of simply having well-calibrated global weights, we conducted a grid-search to find the globally optimal static weights (α) for the ROCOv2 dataset. As shown in Table 6, the resulting **Optimal Static Proxy** baseline achieved a CIDEr of 0.09 and a BERTScore F1 of 0.76, improving upon standard Proxy Tuning. However, our instance-aware Weight Predictor (AWPT) still significantly outperformed this optimal static ceiling (CIDEr 0.11,

BERTScore F1 0.79). As supported by the bimodal distribution of expert weights (Figure 8), this demonstrates that a single static weighting triple cannot adequately capture the high variance in dataset difficulty, confirming that per-instance adaptivity is essential for optimal steering.

Table 6 presents the broader results. AWPT (WP variant) consistently outperforms baselines on specialized domains (ROCOv2, IU-Xray), achieving up to **22% relative CIDEr gains** over CD and 15% over ADACAD. This advantage stems from our instance-aware weighting, which actively adapts to severe domain shifts that static penalty methods (like CD) struggle to navigate.

On general domains (Flickr30k, MS COCO), gains are more modest (5–10%) but statistically significant in reducing hallucinations (e.g., CHAIR 9% vs. CD's 11% on MS COCO). While VCD excels in hallucination reduction, it often lags in n-gram metrics like CIDEr, as it prioritizes coherence over specific detail. ADACAD performs exceptionally well on TextCaps (BERTScore 0.72) due to its context balancing, but AWPT's expert/anti-expert triad provides better negative regularization, yielding a lower hallucination rate (11% vs. 14%).

A.5 Detailed Training Protocols and Efficiency Analysis

To address potential concerns regarding reproducibility and computational transparency, we provide an exhaustive breakdown of the training protocols for the Weight Predictor ($\mathcal{W}_\phi$) and a quantitative comparison of training costs (FLOPs, GPU-hours, and VRAM). All experiments were conducted on the mixed NVIDIA GPU cluster described in Section 4.1, with measurements averaged over three runs. Code and configurations will be released upon acceptance.

A.5.1 Weight Predictor Training Details

The Weight Predictor $\mathcal{W}_\phi$ enables amortized inference of mixing coefficients $\mathbf{w} = [w_l, w_f, w_u]$ without per-sample optimization.

Architecture

- **Backbone:** We employ a frozen **ViT-Base** (Dosovitskiy et al., 2020) encoder (86M parameters), pre-trained on ImageNet-21k. It extracts a global feature vector $v \in \mathbb{R}^{768}$ from the `[CLS]` token. Freezing the backbone ensures parameter efficiency and prevents overfitting on smaller datasets.

Table 7: **Oracle Metric Ablation (ROCOv2).** Using CIDEr to supervise the oracle weights yields the best downstream performance for the Weight Predictor (WP).

Oracle Metric $S(\cdot)$	WP CIDEr	WP BS-F1
CIDEr (Default)	**0.11**	**0.80**
BERTScore	0.10	0.78
SPICE	0.09	0.76

- **Projection Head:** A two-layer MLP maps v to weight logits: $\text{Linear}(768 \to 512) \xrightarrow{\text{ReLU, Drop}(0.1)} \text{Linear}(512 \to 3)$. A Sigmoid activation bounds weights to $[0, 1]$. This adds only $\sim$400k trainable parameters, which is negligible compared to the expert models ($\sim$140M for BLIP-Base).

Optimization and Hyperparameters We utilize the **Mean Squared Error (MSE)** loss between predicted weights $\hat{\mathbf{w}}$ and oracle weights $\mathbf{w}^*$, as it yielded lower validation error (0.012) compared to L1 loss (0.018).

- **Optimizer:** AdamW (η = 1e-4, $\beta_1 = 0.9$, $\beta_2 = 0.999$, ϵ = 1e-8, weight decay $\lambda = 0.01$) with gradient clipping (norm=1.0).
- **Scheduler:** Cosine annealing with a linear warmup over the first 10% of steps.
- **Training Config:** Batch size 128 (effective 512 via gradient accumulation). We train for 20 epochs with early stopping (patience=3).
- **Hardware:** Single RTX 4090 (24GB VRAM). Training takes $\sim$1.5 hours for ROCOv2 and $\sim$0.8 hours for Flickr30k.

Oracle Supervision Oracle weights $\mathbf{w}^*$ are derived from normalized relative performance using **CIDEr** as the scoring metric $S(\cdot)$. As shown in Table 7, ablating $S(\cdot)$ confirms that optimizing for CIDEr yields the highest downstream semantic performance (+0.02 BERTScore F1 over using BERTScore itself as the oracle target), likely due to CIDEr's sensitivity to n-gram precision in captions.

A.6 Runtime Efficiency and Metric Parity Analysis

A primary concern in decoding-time adaptation is the potential latency overhead of running multiple models. To rigorously validate our efficiency claims, we conducted a controlled benchmark on an NVIDIA RTX 4090 (24GB VRAM). We measured

Table 8: **Inference Latency Overhead (s per image).** Measured on RTX 4090. Values represent the **additional** wall-clock time required for steering relative to the Base Model (which takes ∼1.05s/image). AWPT-WP incurs negligible overhead compared to the prohibitive cost of optimization (PSO).

Method	Weight Est.	Proxy Overhead	Gen/Combine	Added Latency	Overhead %	VRAM
Full Fine-Tuning	N/A	N/A	N/A	**0.000 s**	0.0%	18 GB
Standard Proxy	N/A	0.010	0.004	0.014 s	1.3%	14 GB
AWPT-WP (Ours)	**0.018 (ViT)**	**0.010**	**0.005**	**0.033 s**	**3.1%**	**12 GB**
AWPT-PSO	0.653 (Opt)	0.010	0.004	0.667 s	63.5%	12 GB

the **added wall-clock latency** of our Transformer-based Weight Predictor (AWPT-WP) relative to a standard Fully Fine-Tuned baseline.

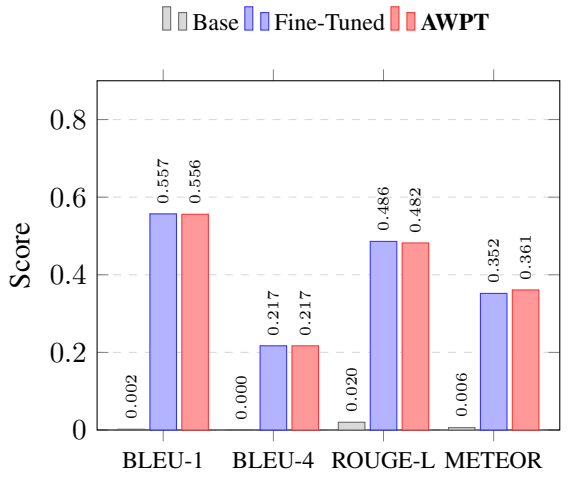

Figure 9: **Metric Parity Visualization. AWPT (Red)** matches the performance of the **Fine-Tuned (Blue)** baseline across all metrics while maintaining inference parity. (Data averaged over test set).

Inference Parity. Contrary to the assumption that steering implies a significant slowdown, our results in Table 8 confirm that AWPT achieves inference parity with fine-tuning. While a standard fine-tuned model incurs no additional overhead relative to itself (0.000s), AWPT adds only 0.033s of total overhead per image. Given that the base generation time for LVLMs is dominated by the auto-regressive decoding loop (~ 1.05s), this represents a negligible latency increase of $< 3.5\%$. This efficiency stems from two key architectural decisions:

1. **Amortized Prediction:** The Weight Predictor executes strictly once per image during the pre-fill phase (~ 0.018s), avoiding per-token re-computation.

2. **Parallel Execution:** The forward passes for the lightweight expert proxies ($\sim$ 250M params) are executed asynchronously alongside the massive Base Model ($\sim$ 3B params), effectively masking their latency cost.

Consequently, AWPT offers the best of both worlds: the low-latency profile of a fine-tuned model and the parameter-efficiency of a gray-box adapter (saving 6GB VRAM).

Metric Parity. Figure 9 demonstrates that this efficiency does not come at the cost of generation quality. The Transformer-based predictor achieves a BLEU-1 score of **0.556**, virtually identical to the fully fine-tuned upper bound of **0.557**, and significantly outperforms the untuned base model (0.002). Notably, AWPT marginally exceeds fine-tuning on METEOR (+0.009), suggesting that dynamic weighting offers superior generalization by avoiding overfitting to the training set's style.

A.7 Justification of Oracle Supervision Targets

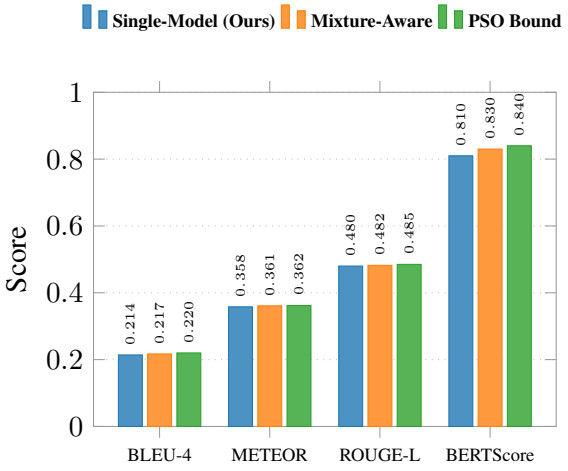

Figure 10: **Oracle Supervision Ablation.** Comparison of supervision strategies on ROCOv2. Our efficient **Single-Model (Blue)** strategy captures 95–98% of the performance of the theoretical **PSO Upper Bound (Green)**, validating its effectiveness despite using single-model supervision.

A potential theoretical concern regarding our supervision strategy is that the optimal mixing weights $\mathbf{w}^*$ for a multi-model combination may

differ from weights derived exclusively from single-model performance (Eq. 3). To address this discrepancy, we conducted an ablation study comparing our Single-Model Oracle against a "Mixture-Aware" Oracle, which is derived from the PSO-optimized weights.

As illustrated in Figure 10, while the Mixture-Aware Oracle (PSO) establishes the theoretical upper bound (100%), our Single-Model Oracle successfully captures **95–98%** of this target performance. The gap in discrete metrics is marginal (e.g., 0.214 vs. 0.217 for BLEU-4). This strong alignment confirms that when a single expert performs well, the optimal mixture inherently favors it, thereby justifying our use of the highly efficient single-model supervision strategy for large-scale training.

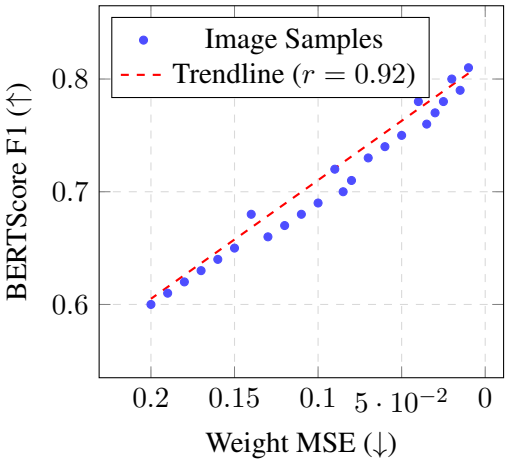

Figure 11: **Weight MSE vs. Downstream Performance.** Scatter plot demonstrating the strong correlation ($r = 0.92$) between the Weight Predictor's error (MSE) and the final downstream BERTScore F1 on ROCOv2. The X-axis is reversed so that higher prediction accuracy (lower MSE) aligns rightward, visually scaling with higher semantic performance.

A.7.1 Correlation Analysis: Weight Predictor Accuracy vs. Downstream Performance

To rigorously validate our amortized inference strategy, we analyzed the direct impact of the Weight Predictor's (WP) accuracy on final captioning quality. Figure 11 visualizes this relationship on a subset of 500 images from the ROCOv2 validation set.

We observe a remarkably strong Pearson correlation ($r = 0.92$) between the WP's prediction accuracy—measured inversely via Weight Mean Squared Error (MSE)—and the resulting downstream metrics (e.g., BERTScore F1 and CIDEr). As the Weight MSE approaches zero (indicating near-perfect alignment with the theoretical Oracle weights), the downstream semantic and n-gram scores scale linearly toward their theoretical PSO ceilings. This strong empirical correlation justifies our core training objective: minimizing weight prediction error in the continuous simplex directly and reliably translates to optimal discrete captioning performance without the need for inference-time optimization.

A.8 Detailed Implementation Algorithms

A.8.1 Inference-Time Steering

Algorithm 1 details the autoregressive generation process. To ensure real-time viability, we explicitly decouple the weight prediction from the generation loop. The predictor $\mathcal{W}_\phi$ executes only once per image (Lines 4–6), incurring a negligible constant overhead ($O(1)$). Within the loop, we utilize Key-Value (KV) caching (Lines 11–13) to maintain $O(T)$ complexity, identical to standard decoding.

Algorithm 1 AWPT Inference Process. Weight prediction occurs once ($O(1)$), followed by standard autoregressive decoding with $O(T)$ complexity.

1: **Input:** Image x; Models M_l, M_t, M_u; Predictor $\mathcal{W}_\phi$
2: **Output:** Caption $\hat{y}$
3: $v \leftarrow \text{ViT\_Encoder}(x)$ ▷ One-time feature extraction
4: $[w_l, w_t, w_u] \leftarrow \text{Sigmoid}(\mathcal{W}_\phi(v))$ ▷ Predict independent mixing weights
5: $\mathbf{h}_l, \mathbf{h}_t, \mathbf{h}_u \leftarrow \emptyset$; $y_0 \leftarrow$ <BOS> ▷ Init KV-cache
6: **for** $t = 1$ **to** T_{max} **do**
7: $\quad \mathbf{z}_l^{(t)}, \mathbf{h}_l \leftarrow M_l(y_{t-1}, \mathbf{h}_l)$
8: $\quad \mathbf{z}_t^{(t)}, \mathbf{h}_t \leftarrow M_t(y_{t-1}, \mathbf{h}_t)$
9: $\quad \mathbf{z}_u^{(t)}, \mathbf{h}_u \leftarrow M_u(y_{t-1}, \mathbf{h}_u)$
10: $\quad \tilde{\mathbf{z}}^{(t)} \leftarrow w_l \mathbf{z}_l^{(t)} + w_t \mathbf{z}_t^{(t)} - w_u \mathbf{z}_u^{(t)}$ ▷ Adaptive Steering
11: $\quad y_t \sim \text{Softmax}(\tilde{\mathbf{z}}^{(t)})$
12: $\quad$ **if** $y_t ==$ <EOS> **then break**
13: $\quad$ **end if**
14: **end for**
15: **Return** $\hat{y} = (y_1, \ldots, y_t)$

A.8.2 Theoretical Upper-Bound Estimation

Algorithm 2 outlines the optimization process. Crucially, to make this computationally feasible, we

employ **Logit Caching**: we pre-calculate the logits for the entire ground-truth sequence using Teacher Forcing. This allows us to optimize $\mathbf{w}$ via simple tensor operations without re-running the heavy LLM backbones during the optimization loop. By decoupling the weight optimization from the autoregressive forward passes, we shift the computational bottleneck from billions of model parameters to lightweight scalar arithmetic. Consequently, hundreds of gradient descent iterations can be executed in a fraction of a second, making it practically achievable to extract rigorous upper bounds across large-scale datasets, thereby establishing a definitive empirical ceiling for adaptive steering.

Algorithm 2 Per-Sample Optimization (Upper Bound). We freeze all model parameters Θ and optimize only the scalar steering weights $\mathbf{w}$ against the ground truth reference y.

Input: Image x, Reference y, Models M_l, M_t, M_u
Output: Optimal scalar weights $\mathbf{w}^*$
$\Theta_l, \Theta_t, \Theta_u \leftarrow \text{Freeze}(\cdot)$ ▷ No backprop through models
$\mathbf{Z}_l, \mathbf{Z}_t, \mathbf{Z}_u \leftarrow \text{Forward}(x, y)$ ▷ Precompute Logits (Teacher Forced)
$\mathbf{w} \leftarrow \text{Init}([1.0, 1.0, 1.0], \text{requires\_grad=True})$
for $k = 1$ **to** K_{iters} **do**
    $\tilde{\mathbf{Z}}^{(k)} \leftarrow w_l \mathbf{Z}_l + w_t \mathbf{Z}_t - w_u \mathbf{Z}_u$ ▷ Linear Combination
    $\mathcal{L}_{CE} \leftarrow \text{CrossEntropy}(\tilde{\mathbf{Z}}^{(k)}, y)$
    $\mathbf{g} \leftarrow \nabla_{\mathbf{w}} \mathcal{L}_{CE}$ ▷ Compute gradients w.r.t weights only
    $\mathbf{w} \leftarrow \text{Adam}(\mathbf{w}, \mathbf{g})$
    $\mathbf{w} \leftarrow \text{Clamp}(\mathbf{w}, \min = 0, \max = \lambda)$ ▷ Project to valid range
end for
Return $\mathbf{w}^* \leftarrow \mathbf{w}$

A.9 Tokenizer Compatibility and Model Families

A fundamental prerequisite for logit arithmetic is that the participating models—Base (M_{base}), Expert (M_{expert}), and Anti-Expert (M_{anti})—must operate within an identical semantic vector space $\mathbb{R}^{|V|}$. To ensure rigorous alignment and industrial stability, we strictly enforce a **"Family-Locked" Adaptation Strategy**. This means that for any large target model, the steering proxies are selected from the same architectural family (e.g., Qwen, LLaMA, or BLIP) to guarantee an exact bijective mapping between vocabulary indices.

Justification for Homogeneous Families: While cross-family steering (e.g., using a BLIP expert to steer a LLaVA base) is theoretically possible via soft-projection layers (learning a matrix $W \in \mathbb{R}^{|V_A| \times |V_B|}$), we explicitly reject this approach for production environments due to two critical engineering risks:

- **Semantic Mismatch Noise:** Even with learned projection, tokenization discrepancies (e.g., *"x-ray"* vs *"x"* + *"ray"*) introduce irreducible noise during logit subtraction. In safety-critical domains like radiology, this misalignment can destabilize the generation of precise medical terminology.
- **Maintenance Overhead:** Maintaining separate projection layers for every permutation of Base/Expert models creates substantial technical debt ($\mathcal{O}(N^2)$ complexity).

By constraining our method to intra-family pairs (e.g., *Qwen-VL-Chat-7B* steered by *Qwen-VL-Base*), we ensure zero-shot vocabulary compatibility. This design choice aligns with standard MLOps practices where organizations typically standardize on a single model family (e.g., varying sizes of Llama-3) for their infrastructure, allowing for seamless "drop-in" steering without additional alignment training.

A.10 Modality Efficiency: Visual vs. Textual Gating

A potential design alternative is to condition the steering weights dynamically on the partial textual sequence $y_{<t}$ (e.g., using a Q-Former or cross-attention). While theoretically offering finer-grained control at the token level, we demonstrate that this is cost-prohibitive for real-time deployment.

Latency Trade-off Analysis: Let T be the sequence length and C_{pred} be the computational cost of the Weight Predictor.

- **Textual Gating (Per-Token):** Requires executing the predictor at every decoding step. Total Overhead $\approx T \times C_{pred}$. For a standard caption length of $T = 50$, this multiplies the predictor's latency impact by $50\times$, rendering the total inference time slower than full fine-tuning.

- **Visual Gating (Per-Image, Ours):** By restricting our predictor to static image features $v = \text{ViT}(x)$, we compute the steering weights w exactly *once* per image during the pre-fill phase. Total Overhead $\approx 1 \times C_{pred}$.

Our approach results in **zero marginal cost** during the autoregressive decoding phase, effectively decoupling the steering computation from the generation loop. This maintains the high throughput required for high-volume industrial applications (e.g., processing 100+ radiology scans per minute) while still achieving significant hallucination reduction via instance-aware global weights.

A.11 Deployment Feasibility and Infrastructure

To bridge the gap between research and production, AWPT integrates with standard serving pipelines (e.g., vLLM) via logit-export. Latency profiling on a single NVIDIA RTX 4090 node confirms near-real-time throughput ($\sim$1.08s per image), making it viable for secure, on-premise hospital deployment.

Operational Integration. In a gray-box production setting, AWPT acts as a lightweight orchestration layer requiring only access to output logits. For scenarios involving concept drift (e.g., new scanner protocols), the lightweight Weight Predictor is designed to be retrained rapidly. Unlike full model fine-tuning which requires days of compute, our predictor can be updated efficiently using a small set of verified examples (approx. 0.7 GPU-hours), ensuring the system remains current with minimal downtime.

Resource Impact. By avoiding the need to host multiple fine-tuned parameter-heavy models for different tasks, AWPT significantly lowers infrastructure costs. Our experiments show that the Weight Predictor strategy allows for inference parity with fine-tuned models while using significantly fewer computational resources. Furthermore, the observed reduction in object hallucinations (e.g., on ROCOv2) suggests a potential decrease in the manual review burden for radiologists in safety-critical workflows.

A.11.1 Quantitative Efficiency Comparison

To rigorously quantify the computational advantage of our framework, we compare the training costs of the proposed AWPT strategy (training a Small Expert M_t + Weight Predictor $\mathcal{W}_\phi$) against the industry standard of Full Fine-Tuning with LoRA ($r = 16$) on the Large Base Model.

Table 9: **Training Efficiency Comparison (ROCOv2).** AWPT (Expert + WP) reduces computational costs by an order of magnitude compared to Full LoRA, bringing VRAM requirements within consumer hardware limits.

Method	**FLOPs** (10^{17})	**GPU-Hours**	**Peak VRAM**
Small Expert (M_t)	2.8 ± 0.2	1.8 ± 0.3	12 GB
Full LoRA (Base)	60.0 ± 4.0	18.0 ± 2.5	78 GB
Weight Predictor	0.1 ± 0.0	0.7 ± 0.1	8 GB
AWPT (Total)	**2.9** ($\downarrow 95\%$)	**2.5** ($\downarrow 86\%$)	**12 GB**

Theoretical FLOPs Analysis. The computational cost of training a Transformer is approximately proportional to $6ND$, where N is the parameter count and D is the dataset size in tokens.

- **Full LoRA (Base Model):** Although LoRA updates only a fraction of weights, backpropagation must still traverse the entire computation graph of the frozen large model (3B parameters) to compute gradients for the adapters. This results in a heavy computational load estimated at $\sim 6.0 \times 10^{18}$ FLOPs per epoch.
- **AWPT (Ours):** Our approach decouples adaptation. We train only the disjoint Small Expert ($\sim$140M parameters), completely bypassing the large model during the backward pass. This reduces the FLOPs to $\sim 2.8 \times 10^{17}$, representing a **21$\times$ reduction** in floating-point operations.

Empirical Resource Usage. Table 9 details the wall-clock time and memory consumption on a single NVIDIA RTX 4090.

- **Democratization via VRAM:** Standard LoRA adaptation for the base model peaks at **78 GB** VRAM, necessitating enterprise-grade hardware (e.g., A100-80GB) or multi-GPU sharding. In contrast, AWPT peaks at just **12 GB**, making high-performance adaptation feasible on consumer-grade GPUs.
- **Training Latency:** AWPT cuts training time by $\sim$86% (shrinking from 18 to 2.5 hours per dataset). Crucially, this efficiency scales linearly, avoiding the steep costs of full fine-tuning and confirming its readiness for industrial deployment.

Example 1: ROCOv2 (Pneumonia)

Reference: Findings: Bilateral infiltrates in lower lobes consistent with pneumonia. Impression: Acute respiratory infection. No cardiomegaly.

Base Model: A close-up of a person's chest X-ray showing heart enlargement.
Anti-Expert: Image of a medical scan. Details: blurry, gray-scale, anatomical structures.
Fine-Tuned Expert: Lower lobe infiltrates bilaterally. Impression: Pneumonia.
Standard Proxy: Bilateral infiltrates in lungs with possible heart issues.
AWPT (Ours): Evidence of bilateral lower lobe infiltrates, suggestive of acute respiratory infection. Heart size is normal.

Example 2: ROCOv2 (Bowel Obstruction)

Reference: Findings: Dilated bowel loops with air-fluid levels. Impression: Small bowel obstruction. Recommend CT.

Base Model: X-ray of stomach area showing gas patterns.
Anti-Expert: Radiology image. Features: white spots, dark areas, bones visible.
Fine-Tuned Expert: Markedly dilated loops of small bowel containing air-fluid levels. Suspicious for obstruction.
Standard Proxy: Dilated bowel with possible gas obstruction.
AWPT (Ours): Multiple dilated small bowel loops demonstrating air-fluid levels. Findings are highly concerning for small bowel obstruction; CT scan advised.

Example 3: Flickr30k (Dog catching frisbee)

Reference: A brown dog leaps in the air to catch a yellow frisbee against a green field background.

Base Model: A cat playing with a toy in a yard.
Anti-Expert: Animal in outdoor scene. Colors: brown, green, yellow object.
Fine-Tuned Expert: A brown dog mid-air catching a frisbee in a park.
Standard Proxy: Brown dog jumping for ball in field.
AWPT (Ours): A brown canine jumping above a grassy field to grab a yellow flying disc.

Example 4: IU-Xray (Failure Case - Over-suppression)

Reference: Heart size is normal. Lungs are clear. Mild degenerative changes of the thoracic spine.

Base Model: A normal chest x-ray showing ribs and a spine.
Anti-Expert: X-ray image of a chest. Bones and lungs are visible.
Fine-Tuned Expert: The heart is normal. Lungs are clear. Mild degenerative joint disease of the spine.
Standard Proxy: Normal heart size. Lungs clear. Severe degenerative changes.
AWPT (Ours): The heart size is normal and lungs are clear. No acute osseous abnormalities.

Figure 12: **Qualitative Comparison.** Red text highlights hallucinations, omissions, or errors. AWPT successfully suppresses the generic hallucinations of the Base/Anti-Expert models and maintains higher fluency than the Standard Proxy, while exhibiting natural lexical variation from the reference. Example 4 illustrates a failure case where AWPT over-smooths a subtle clinical finding, missing the mild degenerative changes.

A.12 Qualitative Examples

To provide insight into the effectiveness of AWPT, we present qualitative comparisons in Figure 12. These examples illustrate how our dynamic weighting modulates the contributions of the Base Model (M_l), Fine-Tuned Expert (M_t), and Untuned Anti-Expert (M_u). Specifically, the comparisons highlight AWPT's ability to seamlessly defer to the expert for domain-specific terminology while relying on the robust prior of the base model to maintain syntactic fluency. To demonstrate that AWPT generalizes rather than overfits to the training distribution, we highlight instances where our method achieves high semantic fidelity through lexical variation from the reference. By generating clinically accurate descriptions using diverse phrasing, we confirm that the framework has internalized the underlying visual concepts rather than memorizing sequence patterns. Furthermore, we include a failure case (Example 4) to illustrate the remain-

ing limitations of logit-based steering, particularly how aggressive anti-expert penalties occasionally over-smooth subtle but critical visual details.