AUDITING OPTIMIZATION AND PROBABILISTIC RELIABILITY IN FROZEN VISION-LANGUAGE REPRESENTATION INTERVENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

We study whether optimization fit, probabilistic prediction quality, and discrete behavior move together under scheduled representation intervention in a frozen vision-language model. PAAF organizes the analysis into a deterministic Formation Schedule, normalized token-wise Allocation, and the Action that applies the resulting vector field. We compare the held-out training objective with Accuracy, ECE, Brier score, and NLL. In paired-coordinate KnG evaluations (2×2 design), gain scheduling lowers the held-out training objective by 0.0999 while increasing NLL by 0.0734 and Brier score by 0.0396. A controlled Allocation perturbation changes internal readouts but changes only 3 of 500 decisions. These results show that optimization fit, probabilistic prediction quality, and decision sensitivity are distinct properties under the evaluated intervention configuration. The practical implication is that representation-intervention schedules should be selected using held-out proper scores and decision-level checks, not the training objective alone.

1 INTRODUCTION

While Vision-Language Models (VLMs) excel at image captioning and VQA, they often rely on surface visual cues and struggle with domain-specific knowledge Chen et al. (2019). This limitation stems from overfitting to robust pretraining priors, which suppresses target-domain adaptation under weak signals Ramasesh et al. (2021). Frozen vision-language models can be adapted through representation interventions without updating the backbone. However, a lower optimization objective does not by itself establish better probabilistic predictions or more reliable decisions.

Although frozen VLMs project representations into a shared language backbone, representation intervention enables direct computation modification without updating frozen parameters. However, existing techniques—including knowledge distillation Hinton et al. (2015), knowledge injection Xu et al. (2023), PEFT methods like LoRA Hu et al. (2022) and adapters Hu et al. (2023), and representation-level interventions Wu et al. (2024c)—typically operate at a coarse granularity. They often optimize training objectives without ensuring that optimization fit, probabilistic prediction quality, and decision sensitivity remain aligned, creating an evaluation gap.

To address this, we introduce the Policy–Allocation–Action framework (PAAF), which separates when intervention controls are formed, how their mass is allocated across learned vector fields, and which Action applies the resulting field to the frozen representation. This framework separates three estimands: trajectory response, Allocation response, and Action-dependent output response. The Schedule dictates optimization trajectories for representation gain and Allocation temperature, whereas the Action defines how the resulting vector field modifies frozen representations using components such as VEFT Kawamae (2025c) and KnG Kawamae (2025b). This decomposition enables matched tests of whether improved optimization fit corresponds to improved probabilistic reliability and decision behavior.

This connects representation intervention to two challenges: nonstationary control trajectories cause path-dependent optimization outcomes not captured by time averages, and unchanged Accuracy can mask distinct probability distributions. We thus treat Schedule evaluation as a joint optimization and probabilistic prediction task.

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.

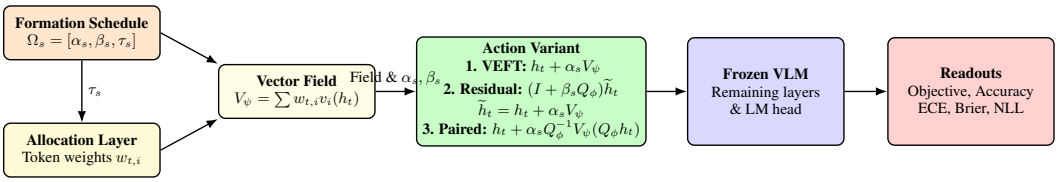


Figure 1: **Schedule–Allocation–Action structure in PAAF.** The Formation Schedule specifies the active gains and Allocation temperature. Allocation forms a token-wise vector field, and the selected Action modifies the frozen representation before the reported objective, probability, and decision readouts.

We test one primary, one secondary, and one diagnostic claim: (1) a Gain-by-Temperature factorization in Paired-coordinate KnG tests which scheduled component is associated with the objective and proper-score shifts; (2) a matched Action-by-Schedule experiment tests whether Schedule response differs across Actions; and (3) a controlled Allocation perturbation measures whether internal changes alter final discrete decisions. These findings establish an evaluation template: match control strength, factorize components, report proper scores alongside Accuracy, and distinguish internal from decision responses.

2 Related Work

Weight-space PEFT updates restricted parameters, whereas representation-intervention methods modify hidden states Hu et al. (2022); Wu et al. (2024c). Conditional activation steering applies steering directions according to input context Lee et al. (2025). PAAF instead studies optimization-time trajectories over intervention gain and Allocation temperature. Concurrent work uses the same frozen-VLM substrate to study evaluation-time transmission from Prompt composition to Allocation, realized actuation, and decisions Anonymous (2026b). PAAF further unifies the VEFT Kawamae (2025c) and KSL Kawamae (2025a) substrates through a common mechanism for coordinating Formation Schedule, Allocation, and Action.

Conditional-computation and routed-adaptation methods assign input-dependent weights across multiple paths Shazeer et al. (2017); Fedus et al. (2022). While approaches like KSL Kawamae (2025a) and MoLE Wu et al. (2024b) apply probabilistic or hierarchical gating over subnetworks or modular components, PAAF does not claim novelty from combining multiple branches with a learned gate. Instead, it investigates matched Schedule contrasts and stage-specific readouts. Softmax and sparsemax Martins & Astudillo (2016) are evaluated as Allocation mappings; their comparison is a manipulation check rather than a methodological contribution.

Hyperparameter schedules alter optimization dynamics even when average values are matched Loshchilov & Hutter (2017). Unlike static steering methods, the Formation Schedule treats intervention strength as an explicit optimization-time trajectory. Our experiments test whether the training objective, probabilistic prediction quality, and decision sensitivity remain aligned under scheduled intervention. Because objective fit does not fully characterize predictive reliability Guo et al. (2017), we evaluate these schedules using probabilistic metrics including Expected Calibration Error (ECE), Brier score Brier (1950), and negative log-likelihood (NLL)—complemented by downstream perturbation tests.

3 Method — Policy–Allocation–Action Framework

3.1 Overview

We investigate representation intervention in frozen VLMs by modifying hidden states at targeted layers during training and inference. Building on the LHFH and KSL formulations Kawamae (2025a), we use *intervention set* to denote the learned subnetwork vector fields together with their normalized token-wise allocation weights. Because frozen foundation models rely on a shared latent space for multimodal interpretation and generation, they provide a direct interface to exam-

ine whether optimization fit, internal adaptation, and external behavior remain aligned. To capture multi-parameter adaptation, we evaluate deterministic trajectories for representation gain, residual gain, and Allocation temperature, defining active trajectories for VEFT, paired-coordinate KnG, and output-residual KnG. The two KnG Actions use the learned coordinate map differently. Output-residual KnG adds a transformed copy of the VEFT-updated state, whereas Paired-coordinate KnG evaluates VEFT in the transformed coordinate system and maps the result back. The confirmatory endpoints are the held-out training objective, Q-Bench2 Accuracy, Brier score, and NLL, with ECE, generation, and system results serving as secondary or exploratory evaluations. Matched controls and factorial interventions compare the effects of gain, Allocation temperature, and Action parameterization on objective, probabilistic, and decision-level readouts.

3.2 Statistical likelihood layer

Let h_t^{base} denote the frozen representation received by the intervention wrapper and let $h_t^{\text{int}} = A(h_t^{\text{base}})$ denote the representation after the selected Action and final normalization. The frozen unembedding produces full-vocabulary logits $\ell_t = W_U h_t^{\text{int}}$. For multiple-choice evaluations, we extract the logits associated with the single-token variants of A, B, C, and D and renormalize them over these four options: $p_t^{\text{MCQ}} = \text{softmax}\big(\ell_{t,A}, \ell_{t,B}, \ell_{t,C}, \ell_{t,D}\big)$. Reliability metrics—including accuracy, confidence, expected calibration error (ECE), Brier score, and negative log-likelihood (NLL)—are evaluated using p_t^{MCQ}.

3.3 Representation dynamics and the Action layer

Let h_t denote the frozen backbone representation and let m_t denote the Prompt-derived controller state used by the allocator. The Allocation layer constructs $V_\psi(h_t, m_t; \tau_s) = \sum_{i=1}^{K} w_{t,i}(h_t, m_t; \tau_s) v_i(h_t)$.

VEFT. $A_{\text{VEFT}}(h_t, m_t; s) = h_t + \alpha_s V_\psi(h_t, m_t; \tau_s)$.

Output-residual KnG. Let $\widetilde{h}_t = h_t + \alpha_s V_\psi(h_t, m_t; \tau_s)$. Output-residual KnG applies $A_{\text{res}}(h_t, m_t; s) = \widetilde{h}_t + \beta_s Q_\phi \widetilde{h}_t$. We write $Q_\phi = D_\phi P$. In reported runs, the permutation is fixed to $P = I$, while $D_\phi = \text{diag}(q_1, \ldots, q_d)$ contains trainable binary coordinate signs. The forward pass uses $q_i \in \{-1, +1\}$ and the backward pass uses a $\tanh$ straight-through surrogate, making the map a learned coordinate-wise sign transformation rather than a dense learned orthogonal transformation. Although Q_ϕ is orthogonal and has $O(d)$ application cost, the complete output-residual map $I + \beta_s Q_\phi$ is not generally orthogonal or norm preserving.

Paired-coordinate KnG. $A_{\text{pair}}(h_t, m_t; s) = Q_\phi^{-1} A_{\text{VEFT}}(Q_\phi h_t, m_t; s) = h_t + \alpha_s Q_\phi^{-1} V_\psi(Q_\phi h_t, m_t; \tau_s)$. This Action serves as a coordinate-conjugated parameterization of the intervention field. Because $V_\psi(Q_\phi h) \neq Q_\phi V_\psi(h)$ need not hold, we do not interpret it as a pure coordinate re-expression or a learned semantic transformation. Global gains α_s and β_s are functions of optimization step s provided by the policy layer.

In our implementation, the operator mixture is applied to the final hidden representation before the frozen language model head. Layer 23 denotes the final text-transformer block, placed adjacent to the language model head. This placement provides a short observation path from the modified representation to the frozen language-model head. The reported effects may depend on the intervention location. Setting $K = 1$, $\beta_s = 0$, and constant α_s yields a single static representation-space operator related to single-operator interventions such as ReFT Wu et al. (2024c). VEFT is not a weight-space PEFT method, and allocation weights $w_{t,i}$represent probabilistic weighting over sub-networks rather than discrete expert selection.

3.4 Allocation layer

The allocation is produced by a router R from the hidden state and a temperature: $w_t = R(h_t, m_t; \tau_s) \in \Delta^{K-1}$, where Δ^{K-1} is the probability simplex over K operators, and τ_s is the

Table 1: Taxonomy of Policy variants used for scheduling and ablation studies.

Variant	Description
`scheduled`	α_s ramps up and τ_s anneals (the proposed trajectory)
`fixed_initial`	gains are fixed at their initial values
`fixed_terminal`	gains are fixed at their terminal values
`fixed_timeavg`	α_s, τ_s, and, for Output-residual KnG, β_s are fixed to their discrete time averages
`gain_only`	α_s follows the scheduled trajectory while τ_s is fixed to its discrete time average; β_s is not used for Paired-coordinate KnG
`temperature_only`	α_s is fixed to its discrete time average while τ_s follows the scheduled trajectory; β_s is not used for Paired-coordinate KnG
`random`	gains are sampled independently within their ranges

optimization-step-dependent temperature from the Policy layer. The router acts as an allocator over operators, not as a transformation operator itself.

We report experiments with softmax and sparsemax. Other mappings are not part of the reported empirical comparison. We summarize an allocation with two quantities:

$$C_t = 1 - \frac{H(w_t)}{\log K}, \quad K_{\mathrm{eff},t} = \exp\big(H(w_t)\big) \tag{1}$$

where $H(w_t)$ denotes the entropy of the allocation vector w_t. Geometrically, w_t is a point on the probability simplex. $C_t = 0$ corresponds to uniform allocation across all operators, whereas $C_t = 1$ corresponds to a vertex of the simplex, where a single operator receives all the allocation weight. $K_{\mathrm{eff},t}$ represents the entropy-based effective support.

3.5 Observation layer

For each configuration, we record an operating point

$$o = \big[\mathrm{Acc}, \mathrm{ECE}, \mathrm{Brier}, \mathrm{NLL}, \mathrm{SBERT}, \mathrm{BERTScore}, \mathrm{BLEU}, \text{gen-length}, \text{early-stop}, \mathrm{runtime}, \mathrm{ms/tok}, \mathrm{VRAM}\big].$$

We distinguish confirmatory objective and Q-Bench2 endpoints from exploratory generation, Allocation, and systems measurements. Confirmatory claims are not based on exploratory geometric proxies.

3.6 Formation Schedule

The deterministic Formation Schedule specifies how intervention controls vary over optimization steps s:

$$\Omega_s^{\mathrm{VEFT}} = [\alpha_s, \tau_s], \quad \Omega_s^{\mathrm{res}} = [\alpha_s, \beta_s, \tau_s], \quad \Omega_s^{\mathrm{pair}} = [\alpha_s, \tau_s],$$

where α_s and β_s scale the intervention gains and τ_s sets the router temperature. The implementation decouples two distinct aspects of this Policy design.

Coupling mode. The coupling mode specifies the relationship between the two intervention gains. For Output-residual KnG only, the coupling mode specifies a heuristic relationship between the VEFT gain and residual gain. We do not claim this coupling is theoretically optimal.

Schedule variant. This ablation compares temporal control trajectories under matched average control values across the implemented Action variants. It does not separate Schedule shape from every path-dependent consequence of training.

The fixed-time-average control serves as the reference condition for the temporal scheduling analysis and is constructed to match the logged mean values of the scheduled variables α_s, β_s, and τ_s. Differences between the two conditions therefore capture optimization-dynamic effects not explained by those matched time averages; they do not isolate a trajectory effect from every other path-dependent consequence of training. These matched averages are verified through per-step logs, showing that

Table 2: Action-by-Schedule experiment. Values are mean ± standard deviation over the matched seeds $\{20250917, 20250918, 20250919\}$. All conditions use softmax Allocation and $K = 5$. The objective is the held-out training objective implemented for the VEFT family.

Schedule	**Objective ↓**	**Q1 ECE ↓**	**Q2 ECE ↓**	**VisIT SBERT ↑**
VEFT				
Fixed	0.5781 ± 0.0045	0.1348 ± 0.0471	0.1534 ± 0.0241	0.4276 ± 0.0025
Scheduled	0.5325 ± 0.0111	0.2439 ± 0.0197	0.2474 ± 0.0284	0.4218 ± 0.0088
Output-residual KnG				
Fixed	0.5637 ± 0.0102	0.1249 ± 0.0432	0.1558 ± 0.0274	0.4219 ± 0.0011
Scheduled	0.5325 ± 0.0143	0.2799 ± 0.0310	0.2807 ± 0.0499	0.4219 ± 0.0026
Paired-coordinate KnG				
Fixed	0.7924 ± 0.0108	0.1061 ± 0.0433	0.1322 ± 0.0228	0.4241 ± 0.0054
Scheduled	0.6817 ± 0.0234	0.1236 ± 0.0224	0.1816 ± 0.0253	0.4290 ± 0.0034

Table 3: Allocation dynamics in the Action–Policy experiment. Values are mean ± standard deviation over three seeds after averaging the per-step statistics within each run. All conditions use the softmax allocator.

Action / Policy	Policy	Entropy	Concentration	Effective K
VEFT				
	Fixed	0.8221 ± 0.1556	0.4892 ± 0.0967	2.3144 ± 0.3630
	Scheduled	0.5092 ± 0.0808	0.6836 ± 0.0502	1.6899 ± 0.1334
Output-residual KnG				
	Fixed	0.7891 ± 0.0904	0.5097 ± 0.0562	2.2292 ± 0.1994
	Scheduled	0.5411 ± 0.1053	0.6638 ± 0.0654	1.7462 ± 0.1828
Paired-coordinate KnG				
	Fixed	0.8602 ± 0.1608	0.4655 ± 0.0999	2.4096 ± 0.3789
	Scheduled	0.7105 ± 0.1132	0.5585 ± 0.0703	2.0845 ± 0.2318

the scheduled and fixed-time-average conditions match to numerical precision across run means for α_s and τ_s, as well as β_s for Output-residual KnG. Additionally, Gain-only and Temperature-only variants are used exclusively within the targeted Paired-coordinate Policy factorization. After each Action, the implementation applies the frozen final normalization before the language-model head. Because this action contains no additive KnG residual, β_s is omitted from the factorization.

3.7 Multi-readout evaluation

The object of study is the mapping from intervention controls to observable model behavior:

$$\mathcal{O}(\pi, R, A) = [\text{Reliability}, \text{Semantic quality}, \text{Generation stability}, \text{Throughput}].$$

We investigate whether Schedule and Allocation changes produce consistent or interacting changes across the recorded readouts. We do not optimize a scalar summary or claim Pareto optimality.

4 Experiments

4.1 Experimental Design

Our experiments evaluate how matched optimization-time schedules and implemented Action parameterizations change training fit, predictive reliability, and generated outputs in a frozen vision-language model. The held-out objective is the implemented VEFT-family training objective evaluated on the held-out split; it includes token-level cross-entropy and the active Allocation regularizers, where we use the same frozen-VLM substrate and matched data construction as related optimization-time studies Anonymous (2026a;b) for controlled cross-study comparison. Unless otherwise stated,

all experiments utilize SmolVLM-Instruct as the frozen backbone. The model exposes hidden states via `output_hidden_states=True`, enabling internal diagnostics while remaining fully compatible with the Hugging Face Transformers framework. Only the adaptation modules are optimized during training. The shared implementation holds the frozen backbone, data construction, and evaluation pipeline fixed across the primary Schedule and Action comparisons. Unless otherwise stated, experiments are performed on a single NVIDIA A100 (40 GB) GPU using `bfloat16` precision. VEFT-family and weight-space baseline runs utilize a 4-bit NF4 backbone with `bfloat16` compute, whereas the LoReFT-style baseline employs a non-quantized `bfloat16` backbone due to its hook-based implementation. Random seeds are fixed for single-seed experiments and averaged over three seeds where reported.

4.2 Evaluation

Behavior is evaluated under two complementary regimes.

- **Q-Bench** Wu et al. (2024a) evaluates multiple-choice visual reasoning through classification accuracy. Predicted options (A–D) are extracted using a regex-based parser to accommodate minor formatting variations.
- **VisIT-Bench** Bitton et al. (2023) evaluates open-ended multimodal generation. We report BLEU Papineni et al. (2002), ROUGE-L Lin (2004), METEOR Lavie & Agarwal (2007), BERTScore Zhang et al. (2020), SBERT cosine similarity Reimers & Gurevych (2019), and Exact Match. BERTScore F1 and SBERT cosine similarity are computed on the 148 examples with non-empty reference outputs. The current analysis uses run-level VisIT-Bench metrics and logged generation-length summaries. Example-level resampling is performed only for runs for which persistent per-example prediction records are available.

  Exact Match is zero for every listed method and is reported only in the appendix. Lexical and semantic metrics are interpreted as complementary reference-based measures rather than direct indicators of reasoning quality.

The proposed internal diagnostics are evaluated independently of behavioral metrics. Routing Policy Stability (RPS) measures the Jensen–Shannon divergence between routing-related score distributions under paired conditions, while Representation Drift quantifies hidden-state changes induced by explanation conditioning. Behavioral metrics therefore provide task-level context, whereas RPS and Representation Drift characterize condition-dependent internal responses.

4.3 Action × Policy multi-seed factorial

This is the primary multi-seed experiment used to evaluate the effects of the Action parameterization and Policy trajectory. We compare VEFT, Output-residual KnG, and Paired-coordinate KnG under scheduled and matched fixed-time-average Policies. Each cell is evaluated with the same three training seeds $\{20250917, 20250918, 20250919\}$. The seed is the independent training-replication unit; the 18 run-level rows form a blocked factorial design rather than 18 independent replications.

Scheduling lowers the held-out training objective for all three Actions. Its reliability response differs across the implemented Actions. In the seed-blocked model, the Q1 ECE interval for the Paired-coordinate-by-Scheduled interaction excludes zero (-0.0917, 95% CI $[-0.1811, -0.0023]$), whereas the corresponding Q2 interval includes zero. The SBERT effects are unresolved and are treated as exploratory.

4.4 Example-level decomposition of reliability cost

The observed degradation in calibration under scheduled policies is not explained by a material net loss in classification accuracy. Across the evaluated conditions, scheduled policies consistently increase prediction confidence while yielding higher Expected Calibration Error (ECE) without a corresponding improvement in accuracy.

The proper-score degradation is accompanied by higher four-choice confidence, while correctness transitions remain nearly balanced.

Table 4: Policy factorization for Paired-coordinate KnG. Results are mean ± standard deviation over three matched seeds. Fixed values are the discrete time averages of the corresponding scheduled trajectories. All conditions use softmax Allocation and $K = 5$ subnetworks.

α_s	τ_s	Eval. Loss	Q1 Acc.	Q1 ECE	Q1 Brier	Q1 NLL	Q2 Acc.	Q2 ECE	Q2 Brier	Q2 NLL	SBERT
Fixed	Fixed	0.7924 ± 0.0108	53.80 ± 2.55	0.1061 ± 0.0433	0.5779 ± 0.0246	1.0164 ± 0.0468	41.07 ± 0.90	0.1322 ± 0.0228	0.7052 ± 0.0064	1.2562 ± 0.0228	0.4241 ± 0.0054
Scheduled	Fixed	0.6925 ± 0.0045	51.80 ± 2.36	0.1627 ± 0.0810	0.6258 ± 0.0619	1.0894 ± 0.1155	40.93 ± 1.94	0.1944 ± 0.0336	0.7449 ± 0.0188	1.3297 ± 0.0407	0.4227 ± 0.0043
Fixed	Scheduled	0.7837 ± 0.0118	52.47 ± 1.15	0.1443 ± 0.0442	0.6055 ± 0.0319	1.0686 ± 0.0678	41.07 ± 0.50	0.1626 ± 0.0278	0.7185 ± 0.0225	1.2926 ± 0.0653	0.4247 ± 0.0037
Scheduled	Scheduled	0.6817 ± 0.0234	53.07 ± 0.42	0.1236 ± 0.0224	0.5968 ± 0.0031	1.0402 ± 0.0156	40.20 ± 1.78	0.1816 ± 0.0253	0.7347 ± 0.0269	1.3111 ± 0.0333	0.4290 ± 0.0034

Table 5: Seed-blocked 2×2 Policy-factorization model for Paired-coordinate KnG. Gain and Temperature denote scheduling of α_s and τ_s, respectively. Fixed–fixed is the reference condition. Intervals are model-based 95% confidence intervals over 12 run-level observations. For each metric, we fit a linear model with Gain, Temperature, their interaction, and seed-block indicators. Intervals are computed from the fitted run-level model.

Metric	Effect	Coefficient	95% CI
Eval. Loss	Gain	-0.0999	$[-0.1256, -0.0742]$
	Temperature	-0.0087	$[-0.0343, 0.0170]$
	Gain × Temperature	-0.0021	$[-0.0384, 0.0342]$
Q2 ECE	Gain	$+0.0621$	$[+0.0137, +0.1106]$
	Temperature	$+0.0304$	$[-0.0180, +0.0788]$
	Gain × Temperature	-0.0431	$[-0.1116, +0.0254]$
Q2 Brier	Gain	$+0.0396$	$[+0.0091, +0.0702]$
	Temperature	$+0.0133$	$[-0.0173, +0.0438]$
	Gain × Temperature	-0.0234	$[-0.0666, +0.0197]$
Q2 NLL	Gain	$+0.0734$	$[+0.0055, +0.1414]$
	Temperature	$+0.0364$	$[-0.0315, +0.1043]$
	Gain × Temperature	-0.0549	$[-0.1510, +0.0411]$
SBERT	Gain	-0.0014	$[-0.0112, +0.0084]$
	Temperature	$+0.0006$	$[-0.0092, +0.0104]$
	Gain × Temperature	$+0.0058$	$[-0.0081, +0.0197]$

4.5 Factorizing gain and Allocation-temperature trajectories

The Action–Policy factorial demonstrates that Paired-coordinate KnG responds distinctly to scheduling compared to VEFT and Output-residual KnG. Because the joint schedule modifies both the VEFT intervention gain α_s and the Allocation temperature τ_s, we factorize these trajectories within Paired-coordinate KnG using a matched three-seed 2×2 design: $S_\alpha \in \{\text{fixed}, \text{scheduled}\}$, $S_\tau \in \{\text{fixed}, \text{scheduled}\}$. The existing fixed-time-average and jointly scheduled runs populate the $(0, 0)$ and $(1, 1)$ cells of this matrix. Gain-only and Temperature-only runs complete the four cells under identical seeds, softmax Allocation, and $K = 5$ subnetworks. Because Paired-coordinate KnG does not use the additive residual gain β_s, this design isolates the individual contributions of α_s and τ_s. The direction of the Gain-only reliability effect remains consistent across all three Q-Bench2 categories. We use this analysis as a heterogeneity check and do not claim statistically distinguishable differences among category-specific effect sizes.

The factorization varies the gain and Allocation-temperature trajectories separately within the evaluated Paired-coordinate configuration. Gain scheduling lowers the implemented held-out objective while worsening the evaluated proper scores in the Paired-coordinate configuration.

Logged control values confirm the intended intervention: gain scheduling lowers the held-out training objective by 0.0999 (from 0.7924 to 0.6925), whereas the mean Temperature-only objective change is smaller and its interval includes zero (0.7837). However, this loss reduction does not translate to downstream reliability, as gain scheduling increases the Q-Bench2 ECE ($+0.0621$), Brier score ($+0.0397$), and NLL ($+0.0734$) while changing accuracy by only -0.0013.

Furthermore, correct-to-incorrect and incorrect-to-correct transitions are nearly balanced at 217 and 215. The proper-score degradation is accompanied by higher four-choice confidence without a net Accuracy gain. Within the evaluated Paired-coordinate configuration, gain scheduling shows the clearest objective and proper-score shifts. The intervals for the Temperature main effect and Gain-

Table 6: Matched Q-Bench2 differences for Gain-only relative to Fixed. The same 500 examples are evaluated by each of three independently trained seed models. Example-level intervals summarize matched-example variability within these trained models and do not replace the run-level seed-blocked analysis.

Metric	Mean difference	95% interval
Accuracy	-0.0013	$[-0.0280,\ 0.0247]$
Confidence	$+0.0622$	$[+0.0541,\ +0.0703]$
Brier score	$+0.0397$	$[+0.0183,\ +0.0598]$
NLL	$+0.0734$	$[+0.0379,\ +0.1082]$

Table 7: Controlled Allocation perturbation on 500 examples under matched evaluation conditions.

Comparison	Routing JSD	Linear CKA	Predictive Effect
Normal vs. Repeat	0.0000	1.0000	Identical probabilities
Normal vs. No-prompt-to-gate	8.22×10^{-6}	0.99999994	Zero prediction changes
Normal vs. Token-shuffle	2.12×10^{-3}	0.99996030	3 / 500 decisions changed

by-Temperature interaction include zero. The experiment therefore does not establish independent behavioral axes. Temperature scheduling produces smaller mean proper-score changes than gain scheduling, with intervals that include zero.

4.6 Matched Reliability and Allocation Sensitivity

We evaluate the same 500 Q-Bench2 examples with each of three trained seed models (Table 6). The example-model rows are matched observations, not 1,500 independent training replications. Gain-only scheduling alters accuracy by only -0.0013 (95% interval $[-0.0280,\ 0.0247]$) relative to the Fixed condition, while increasing mean confidence by $+0.0622$ ($[+0.0541,\ +0.0703]$). The paired differences are $+0.0397$ ($[+0.0183,\ +0.0598]$) in Brier score and $+0.0734$ ($[+0.0379,\ +0.1082]$) in NLL. Correctness transitions remain nearly balanced, with 217 seed-model example evaluations shifting from correct to incorrect and 215 shifting from incorrect to correct. The observed proper-score degradation is accompanied by a higher four-choice confidence distribution without a net Accuracy gain. Because the three trained models are the independent training replications, the example-level intervals are descriptive and are not used for inference about training-seed variability.

We evaluate the response to a cyclic shift of token-wise Allocation vectors under matched evaluation conditions and show its result in Table 7. An exact repeat yields zero routing JSD and no probability shifts. Removing the soft-prompt contribution to the gate induces a negligible JSD (8.22×10^{-6}) and alters no predictions. However, cyclically shuffling token-wise Allocation vectors while preserving their multiset increases routing JSD to 2.12×10^{-3}. The repeat and prompt-removal controls separate deterministic replay from the tested Allocation perturbation. The audit does not establish that dynamic Allocation is behaviorally necessary, robust, redundant, or ineffective outside the evaluated perturbation.

5 Discussion and Limitations

The experiments support claims across three levels. First, gain scheduling yields a lower held-out training objective alongside higher Brier and NLL values without Accuracy gains. Because the reported probabilities are renormalized over the four answer options, they characterize four-choice probabilistic quality rather than full-vocabulary calibration. For representation-intervention research, this implies that gain scheduling is not uniformly harmful; rather, objective-driven selection can favor configurations with degraded held-out proper scores, making probabilistic evaluation an integral part of intervention selection rather than a post hoc diagnostic. ECE and confidence capture this probability shift without identifying its mediator.

Second, matched schedule responses vary across implemented actions. Given three training seeds per cell, this reflects the specific design rather than an architecture-independent law.

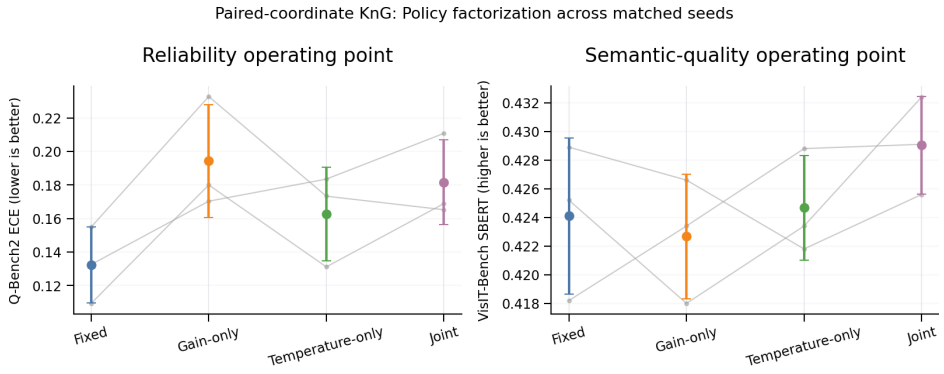


Figure 2: Paired-coordinate Policy factorization across matched seeds. The left panel reports Q-Bench2 ECE and the right panel reports VisIT-Bench SBERT. Gray lines connect the four Policy conditions within each seed; colored points and error bars denote mean $\pm$ standard deviation. Gain scheduling produces the largest increase in Q-Bench2 ECE relative to Fixed, while Joint attains the highest mean SBERT. The seed-blocked SBERT main effects and interaction remain unresolved at three seeds.

Third, cyclic allocation perturbation induces measurable internal changes but limited decision switching, failing to establish robustness, redundancy, or behavioral necessity. Mechanistically, because the intervention applies immediately before the frozen unembedding, the option logits satisfy $\ell = W_U h + \alpha W_U V(h)$ for the direct VEFT term. While a larger gain scales the intervention contribution presented to the unembedding, this relation does not determine whether option margins or confidence increase for a given example. The study leaves open whether probability shifts stem from representation norms, option-logit scales, decision margins, allocation entropy, active regularization, or optimizer dynamics, and the perturbation audit cannot distinguish attenuation, redundancy, weak coupling, or insufficient strength. Confirmatory results are restricted to a single frozen SmolVLM substrate, an intervention placement near the language-model head, and three seeds. External LoReFT and PEFT comparisons serve as contextual references rather than controlled baselines due to differing quantization and implementation paths. Furthermore, the fixed-time-average control matches logged mean controls but does not isolate schedule shape from path-dependent optimization consequences.

Finally, while accuracy remains unchanged, altered confidence distributions impact confidence-based ranking, thresholding, and abstention, though specific deployment policies remain unevaluated.

6 Conclusion

This work introduces PAAF as a framework for forming and evaluating scheduled representation interventions in frozen VLMs. Multi-seed factorial and perturbation analyses show that gain scheduling lowers the implemented held-out objective without improving Accuracy, while increasing ECE, Brier, NLL, and four-choice confidence. Measurable internal shifts do not necessarily produce comparable decision changes. The broader contribution is an evaluation principle: matching average control strength is not enough; scheduled components should be factorized, proper scores should be reported beside Accuracy, and internal response should be distinguished from decision response.

References

Anonymous. Decoding-aware view routing for stable frozen vlm adaptation. In *Under review*, 2026a.

Anonymous. Tracing prompt-conditioned control from allocation to realized actuation in frozen vlms. In *Under review*, 2026b.

Yonatan Bitton, Hritik Bansal, Jack Hessel, Rulin Shao, Wanrong Zhu, Anas Awadalla, Josh Gardner, Rohan Taori, and Ludwig Schmidt. Visit-bench: A dynamic benchmark for evaluating instruction-following vision-and-language models. In *NeurIPS*, 2023.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

Shuang Chen, Jinpeng Wang, Xiaocheng Feng, Feng Jiang, Bing Qin, and Chin-Yew Lin. Enhancing neural data-to-text generation models with external background knowledge. In *EMNLP-IJCNLP*, pp. 3020–3030, 2019.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23:120:1–120:39, 2022.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. volume abs/1503.02531, 2015.

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. In *ICLR*. OpenReview.net, 2022.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *EMNLP*, pp. 5254–5276, 2023.

Noriaki Kawamae. Knowledge-aligned domain shift tuning for efficient adaptation in large language models. In *KDD*, pp. 1128–1138, 2025a.

Noriaki Kawamae. Multi level vision language adapter. In *ICDM*, pp. 1300–1309, 2025b.

Noriaki Kawamae. One-shot multi-view visual conversational recommendation. In *KDD*, pp. 1118–1127, 2025c.

Alon Lavie and Abhaya Agarwal. METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation, WMT@ACL 2007*, pp. 228–231, 2007.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre L. Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *ICLR*, 2025.

Chin-Yew Lin. Rouge: a package for automatic evaluation of summaries. In *ACL-workshop*, pp. 25–26, 2004.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *ICLR*, 2017.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

André F. T. Martins and Ramón Fernandez Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *ICML*, pp. 1614–1623, 2016.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pp. 311–318, 2002.

Vinay Venkatesh Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *ICLR*, 2021.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP*, 2019.

Table 8: Experimental Setup and Baseline Specifications

Model / Baseline	Backbone Precision & Quantization	Hardware Configuration
VEFT-Family & Weight-Space	4-bit NF4 with `bfloat16` compute	Single NVIDIA A100-SXM4-40GB
LoReFT-Style Baseline	Non-quantized `bfloat16`	Single NVIDIA A100-SXM4-40GB
General Experiments	SmolVLM-Instruct (`bfloat16`)	Single NVIDIA A100 (40 GB)

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017.

Haoning Wu, Zicheng Zhang, Erli Zhang, Chaofeng Chen, Liang Liao, Annan Wang, Chunyi Li, Wenxiu Sun, Qiong Yan, Guangtao Zhai, and Weisi Lin. Q-bench: A benchmark for general-purpose foundation models on low-level vision. In *ICLR*, 2024a.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of loRA experts. In *ICLR*, 2024b.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Reft: Representation finetuning for language models. In *NeurIPS*, pp. 63908–63962, 2024c.

Yan Xu, Mahdi Namazifar, Devamanyu Hazarika, Aishwarya Padmakumar, Yang Liu, and Dilek Hakkani-Tür. KILM: knowledge injection into encoder-decoder language models. In *ACL*, pp. 5013–5035, 2023.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *ACL Volume 2: Short Papers*. ACL, 2022.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *ICLR*. OpenReview.net, 2023.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *ICLR*, 2020.

7 AI USE STATEMENT

We used AI tools to edit the manuscript for grammar and readability, assist with literature searches and identify relevant literature, format references, and edit and format tables. All reported results were produced by the authors' experimental pipeline. We manually verified the cited literature, numerical results, code changes, and final claims.

A IMPLEMENTATION DETAILS AND EXPERIMENTAL SETUP

A.1 IMPLEMENTATION AND HARDWARE

Experiments are implemented using PyTorch 2.5.1+cu124 and Hugging Face Transformers. Unless otherwise noted, runs use SDPA attention, bfloat16 compute, and the AdamW optimizer with a linear warm-up and cosine decay schedule (learning rate 2×10^{-4}). Training utilizes a per-device batch size of 1, gradient accumulation of 4, and a single NVIDIA A100-SXM4-40GB GPU. All methods share a unified experimental framework and backbone architecture to ensure consistency.

A.2 TRAINING DATA

To study explanation-conditioned reasoning rather than generic image captioning, training datasets are constructed from reasoning-oriented multimodal examples. Each pair contains both answers and accompanying rationales, enabling a direct comparison between answer-only and answer-plus-explanation supervision under identical inputs.

Table 9: Effect of Policy scheduling and router sparsity. Results are averaged over three random seeds (20250917-20250919). Higher SBERT is better; lower Q2 ECE is better.

Router	Policy	SBERT ↑	Q2 ECE ↓	Gen. Len	Early Stop
Softmax	Fixed	0.4159 ± 0.0010	0.1823 ± 0.0317	64.0 ± 0.0	0.000 ± 0.000
Softmax	Scheduled	0.4274 ± 0.0039	0.2386 ± 0.1012	59.1 ± 6.0	0.133 ± 0.153
Sparsemax	Fixed	0.4179 ± 0.0009	0.2089 ± 0.0525	61.6 ± 2.1	0.067 ± 0.058
Sparsemax	Scheduled	0.4275 ± 0.0014	0.2281 ± 0.0744	62.7 ± 2.3	0.033 ± 0.058

Table 10: Allocation statistics in the earlier Policy–Router experiment. Values are mean $\pm$ standard deviation over three seeds. Concentration is $1 - H/\log K$ and effective operators are $\exp(H)$ with $K = 5$.

Router	Policy	Entropy	Concentration	Effective K
Softmax	Fixed	0.8950 ± 0.1125	0.4439 ± 0.0699	2.4786 ± 0.2785
Softmax	Scheduled	0.6676 ± 0.1119	0.5852 ± 0.0695	1.9793 ± 0.2160
Sparsemax	Fixed	0.1855 ± 0.0813	0.8847 ± 0.0505	1.2095 ± 0.0985
Sparsemax	Scheduled	0.1919 ± 0.1422	0.8808 ± 0.0883	1.2235 ± 0.1779

A.3 Baselines and Adaptation Framework

We compare our approach against representative Parameter-Efficient Fine-Tuning (PEFT) methods from the Hugging Face PEFT library Mangrulkar et al. (2022), including LoRA Hu et al. (2022), AdaLoRA Zhang et al. (2023), IA^3, and BitFit Zaken et al. (2022). VEFT Kawamae (2025c) and KSL Kawamae (2025a) serve as the primary adaptation substrate due to their token-dependent composition weights. Input-level prompt tuning and full-parameter fine-tuning are excluded as primary baselines since they alter the input embedding pathway or violate the frozen-backbone assumption.

A.4 Key Components and Modules

- **Knowledge Gate (KnG):** Implemented as a lightweight wrapper applying a signed permutation matrix to token embeddings via efficient batch matrix operations and block-wise self-inverse orthogonal structures ($\mathbf{W}_{\text{KnG}}$).
- **VEFTKnGScheduler:** Manages policy variables after each optimization step (α_s and τ_s for VEFT and Paired-coordinate KnG; β_s for Output-residual KnG). Entropy feedback is disabled across all experiments.
- **Prompt Injection:** Role-aware tokens (e.g., `<vis_ctx>`, `<question>`, `<answer>`) are prepended, initialized from base embeddings, and jointly trained.
- **Loss Computation:** Token-level losses apply attention masks to exclude padding and prompt-only tokens, incorporating consistency loss and role-aware weights when enabled.
- **Reproducibility:** Experiments use explicit random seeds, standardized command-line logging, and execution metadata to ensure run-level reconstruction.

B Supplementary Results

This section reports implementation settings and secondary or contextual results. These results are not used as the primary evidence for the PAAF Schedule and Action claims.

The router comparison serves as a manipulation check: changing the mapping from softmax to sparsemax changes Allocation entropy, concentration, and entropy-based effective support as intended. Sparsemax produces substantially lower entropy and an effective support close to one weighted subnetwork, whereas softmax retains approximately two weighted subnetworks on average. This concentration is not treated as collapse; it is a valid probabilistic allocation outcome. The router mapping alters the topology of the Allocation distribution as intended. However, these statistics do not establish semantic specialization of individual subnetworks or universal behavioral superiority for sparse Allocation, and the Action–Policy interaction has only been evaluated with softmax.

Table 11: Seed-blocked Action–Policy model. VEFT under the fixed-time-average Policy is the reference condition. Confidence intervals are model-based 95% intervals over the 18 run-level observations.

Metric	Effect	Coefficient	95% CI
SBERT	Scheduled	-0.0058	$[-0.0148,\ 0.0032]$
	Output-residual × Scheduled	$+0.0057$	$[-0.0070,\ 0.0184]$
	Paired-coordinate × Scheduled	$+0.0107$	$[-0.0020,\ 0.0234]$
Q1 ECE	Scheduled	$+0.1091$	$[+0.0459,\ +0.1723]$
	Output-residual × Scheduled	$+0.0459$	$[-0.0435,\ +0.1353]$
	Paired-coordinate × Scheduled	-0.0917	$[-0.1811,\ -0.0023]$
Q2 ECE	Scheduled	$+0.0940$	$[+0.0398,\ +0.1482]$
	Output-residual × Scheduled	$+0.0309$	$[-0.0457,\ +0.1075]$
	Paired-coordinate × Scheduled	-0.0446	$[-0.1212,\ +0.0320]$

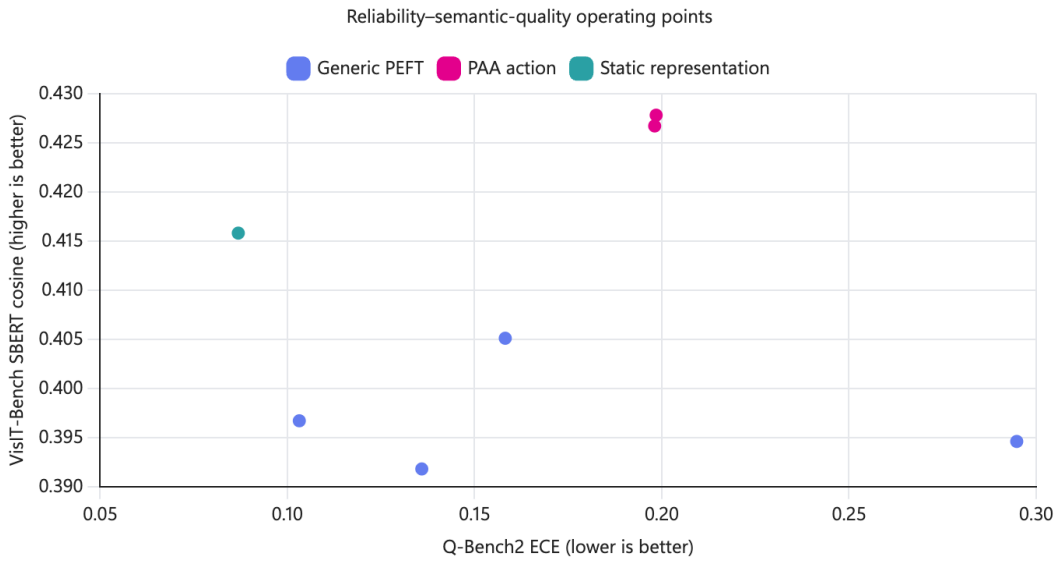


Figure 3: Reliability–semantic-quality operating points for the representative seed 20250917. The horizontal axis reports Q-Bench2 ECE, where lower is better, and the vertical axis reports VisIT-Bench SBERT, where higher is better. The plot is a contextual single-seed comparison rather than a confirmatory multi-seed ranking.

B.1 LoReFT-style placement and rank

We sweep the intervention layer and rank of the static LoReFT-style baseline. The text backbone exposed by the evaluated SmolVLM-Instruct checkpoint contains 24 indexed transformer blocks (0–23) under the resolved Hugging Face model path. Layer 23 is therefore the final usable text block in this implementation.

Validation loss fails to predict downstream quality, as the configuration with the lowest evaluation loss (layer 8, rank 8) yields the weakest Q2 accuracy and SBERT scores, whereas layer 23, rank 8 incurs a higher loss but achieves substantially superior Q2 accuracy and SBERT metrics, with the highest overall Q2 accuracy reached by layer 23, rank 4. Because validation loss rankings diverge from calibrated multiple-choice and semantic metric rankings, we report an operating point rather than selecting based on loss. Across placement configurations, the early block performs weakest downstream, the middle-late block optimizes single-image calibration, and the final block yields the best Q-Bench2, Brier, NLL, and SBERT results. These performance gains are not attributed to language model head proximity; instead, the present placement sweep does not identify the cause

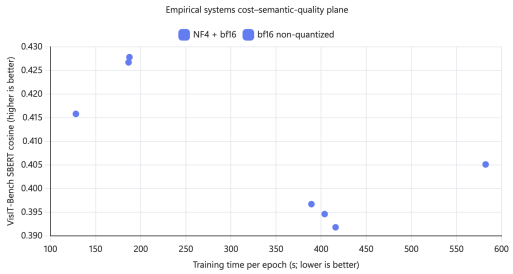


Figure 4: Empirical systems cost semantic-quality plane: Training time per epoch is shown against VisIT-Bench SBERT for the representative seed 20250917. Lower training time and higher SBERT place a method closer to the upper-left region. Marker type distinguishes non-quantized bf16 from NF4 with bf16 compute. Because precision, quantization, adapter topology, optimizer state, and placement are not jointly controlled, the plot is descriptive and does not identify a causal effect of output-side intervention placement.

Table 12: LoReFT-style hyperparameter sweep over intervention layer and rank. All experiments use seed 20250917 and 500 multiple-choice evaluation examples.

Layer	Rank	Trainable Params	Train Loss	Eval. Loss	Q-Bench		Q-Bench2		VisIT SBERT
					Acc	ECE	Acc	ECE	
8	4	20,480	1.066	0.747	48.6	0.134	35.0	0.148	0.4043
8	8	36,864	0.883	0.636	50.0	0.158	31.0	0.213	0.4040
16	4	20,480	1.351	1.013	56.4	0.052	43.8	0.087	0.4108
16	8	36,864	1.149	0.867	58.8	0.057	42.0	0.116	0.4115
23	4	20,480	1.951	1.677	58.0	0.058	46.0	0.089	0.4127
23	8	36,864	1.712	1.367	58.8	0.072	45.2	0.087	0.4158

of these differences. Layer 23, rank 8 is adopted as the representative LoReFT-style configuration in Table 14 based on its overall profile, with confirmatory evaluation across three seeds. Broadly, frozen-model representation adaptation cannot be summarized by a single notion of strength, as optimization-time schedules can lower the implemented held-out objective while worsening proper-score performance, and internally active subnetwork allocation changes can be strongly buffered before reaching discrete behavior. Reflecting this, optimization schedules alter allocation trajectories; under the softmax allocator, routing entropy decreases across all actions, reducing the effective number of weighted subnetworks from 2.31 to 1.69 for VEFT, from 2.23 to 1.75 for output-residual KnG, and from 2.41 to 2.08 for paired-coordinate KnG. This scheduled policy functions beyond a scalar residual-gain schedule by redistributing probability mass over subnetworks through τ_s.

B.2 Throughput

VEFT_only and KnG+VEFT complete training 2.1–3.1× faster than weight-space PEFT baselines despite larger parameter counts. Because methods differ across adapter topologies, optimizers, and quantization (non-quantized bf16 for LoReFT-style), we treat these as empirical system observations rather than attributing gains solely to output-side placement.

B.3 Generic external baselines

We separate the comparison into three tables because reliability, open-ended generation quality, and systems cost answer distinct questions. Table1 reports calibrated MCQ reliability, Table2 reports open-ended generation quality, and Table3 reports empirical systems cost. Tables 14 and 15 provide single-seed contextual comparisons using seed 20250917. They characterize the external operating-point landscape but are not used as the confirmatory multi-seed test of the proposed Policy. The primary multi-seed evidence is reported in the Action–Policy factorial and the Paired-coordinate Policy-factorization experiment. The earlier Policy–Router experiment is retained as a secondary analysis of the Allocation mapping.

Table 13: Training and inference cost comparison. All results use seed 20250917 under the fixed Policy.

Method	Training Time (s)	Latency (ms/token)	Throughput (tokens/s)	Precision
LoReFT-style (23,8)†	128.0	66.8	15.0	bf16 (non-quantized)
VEFT	180.3	98.5	10.2	4-bit NF4 + bf16
Output-residual KnG	178.3	99.1	10.1	4-bit NF4 + bf16
Paired-coordinate KnG	178.1	97.8	10.2	4-bit NF4 + bf16
BitFit	389.2	92.8	10.8	4-bit NF4 + bf16
LoRA	403.9	102.0	9.8	4-bit NF4 + bf16
IA^3	416.0	130.5	7.7	4-bit NF4 + bf16
AdaLoRA	582.4	120.0	8.3	4-bit NF4 + bf16

Table 14: Reliability-oriented evaluation on Q-Bench and Q-Bench2. All results use seed 20250917. The LoReFT-style (23,8) configuration was selected from the sweep reported in Table 12. VEFT-family rows use the scheduled Policy. Multi-seed Action-by-Schedule results are reported in Table 2.

Method	Q1 Acc ↑	Q1 ECE ↓	Q2 Acc ↑	Q2 ECE ↓
Zero-shot	56.8	0.0520	39.8	0.1230
BitFit	56.8	0.0355	40.4	0.1032
LoRA	56.8	0.1814	38.2	0.2948
IA^3	59.8	0.0310	38.6	0.1359
AdaLoRA	57.0	0.0671	37.6	0.1582
LoReFT-style (23,8)†	58.8	0.0723	45.2	0.0869
VEFT	48.2	0.2288	41.2	0.2360
Output-residual KnG	47.0	0.2989	40.8	0.3158
Paired-coordinate KnG	52.6	0.1457	41.6	0.1689

† Selected after the six-configuration sweep in Table 12.

Table 15: Reference-based generation metrics on the 148 VisIT-Bench examples for which the loaded dataset revision exposes non-empty reference outputs. These results should not be interpreted as full-benchmark VisIT-Bench performance. VEFT-family rows employ implementation under the scheduled Policy.

Method	BLEU	ROUGE-L	METEOR	BERTScore	SBERT
BitFit	1.86	12.36	12.93	81.94	0.3967
LoRA	1.57	9.45	13.60	81.52	0.3946
IA^3	2.05	11.28	12.50	81.93	0.3918
AdaLoRA	1.97	12.21	13.00	82.00	0.4051
LoReFT-style (23,8)†	1.98	12.56	14.50	82.19	0.4158
VEFT	2.29	12.49	17.34	83.11	0.4192
Output-residual KnG	2.45	13.15	17.44	83.16	0.4207
Paired-coordinate KnG	2.43	12.78	16.49	83.05	0.4324

† Selected after the sweep reported in Table 12.