TRAIN WIDE, ACT NARROW: THE LEARNING GEOMETRY OF VISUAL ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In video-language adaptation, wide visual adapters learn corrections concentrated almost entirely along one output direction, while their hidden features and readout weights remain broad. Across the compression methods we study, retaining that direction preserves nearly all of the average benchmark gain, while removing it returns accuracy close to the unadapted model. We call this pattern *Train Wide, Act Narrow*. This concentration emerges through learned hidden–readout alignment, with the readout preferentially amplifying a dominant mode of hidden variation. A one-hidden-unit adapter constructed from a trained Wide adapter reaches development negative log-likelihood (NLL) of 1.200, while the best task-trained counterpart we test reaches 1.403. We then compare two readouts that compute exactly the same one-axis function from the same hidden features. The unrestricted readout can produce transverse changes using coefficient patterns unavailable to the one-axis readout. Even the best single pattern misses 17.3 to 32.1% of the transverse gradient energy accessible to the unrestricted readout. After continuation from the same one-axis function, projecting the resulting unrestricted correction onto its dominant direction raises NLL despite retaining more than 99.5% of its raw energy. A nearly one-dimensional correction can require more freedom to learn and more than one direction to preserve its behavior.

1 INTRODUCTION

Increasing hidden width improves fitting in video-language adaptation, yet the learned correction concentrates around one dominant output direction. Across the tested compression methods, keeping that direction preserves nearly all of the average benchmark gain, while removing it returns accuracy close to the unadapted model. The concentration also appears with uncompressed inputs, in another model family, and after training from broad random output weights. We call this pattern *Train Wide, Act Narrow*. We distinguish the dimension of the realized correction, the local learning space around an exactly matched function, and the number of directions needed to preserve fitted behavior. A one-hidden-unit adapter constructed from a trained Wide adapter reaches development negative log-likelihood (NLL) of 1.200, while task training of the same architecture reaches 1.403 at best across four learning rates. A compact architecture can therefore represent a strong correction without readily learning it.

Video-language models (Bai et al., 2025; Chen et al., 2024) encode videos into many visual tokens. Compression methods reduce the token budget but can lose information needed for prediction (Yang et al., 2025b; Shen et al., 2026). We place a residual adapter between the visual projector and the decoder, where it modifies the surviving tokens while the backbone remains frozen. Its correction can be measured, projected, or removed without changing the input condition or token budget. We compare uncompressed inputs with four visual-token compression methods.

This concentrated output does not arise because hidden features or readout weights are themselves narrow. At the default Wide endpoint, both remain broad while the readout amplifies a prominent mode of hidden variation. Rotating the readout relative to the hidden features weakens concentration even when the spectra of both factors are preserved. The concentration therefore depends on learned hidden–readout alignment rather than narrowness of either factor alone.

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.

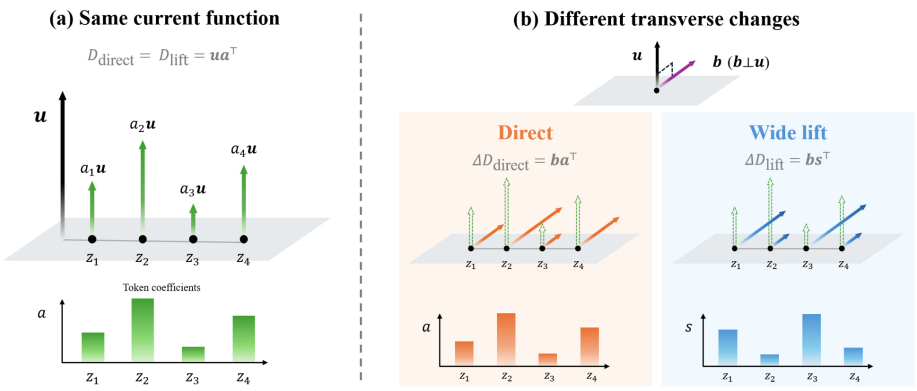

Figure 1. **Same correction, different local learning spaces.** (a) Identical corrections from shared hidden features. (b) Along $b \perp u$, Direct uses the current pattern a, while the Wide lift can use another feature-supported pattern s.

The one-unit comparison tests whether narrow training can find the concentrated correction. We construct a one-hidden-unit SiLU adapter from Wide without further task-loss optimization. Across three Wide runs, it retains about 90% of the development NLL improvement over Base. The same compact architecture expresses a strong correction that direct task training does not find under the tested settings.

We separate hidden width from readout freedom using *Direct*. It retains wide hidden computation but learns a single output axis, reaching generated-answer accuracy similar to Wide at the shared learning rate. We then pair each one-axis function with an unrestricted *Wide lift* that uses the same hidden features and produces exactly the same correction on every input. Direct can rotate the output axis, but at first order it applies the current token coefficient pattern to motion outside that axis. The Wide lift can use other coefficient patterns supported by the same hidden features. At directly trained Direct endpoints, the best single pattern misses 17.3 to 18.9% of the transverse gradient energy available to the Wide lift. The loss gradient with respect to their shared correction is identical for both readouts, so this difference comes from the readout structure, as illustrated in Figure 1.

Continuation from the same one-axis function shows that this extra freedom can affect fitting. With hidden features fixed and the data order shared, the Wide lift reaches lower development NLL than Direct after 500 updates. Projecting each final Wide lift onto its dominant direction removes the point-estimate advantage over Direct, even though that direction carries more than 99.5% of raw correction energy. The original Wide endpoints show a related distinction. One direction preserves average benchmark accuracy but can change individual predictions, while retaining more directions better preserves NLL and prediction agreement. The energy retained by a projection is therefore insufficient to judge whether it preserves fitted behavior.

Prior work studies compact weight updates and intrinsic training dimensions (Hu et al., 2021; Aghajanyan et al., 2021; Li et al., 2018). Representation finetuning uses learned interventions in frozen models (Wu et al., 2024b). Visual steering also trains compact interventions for multimodal tuning (Bi et al., 2025). LoRA-Pro examines optimization of low-rank weight updates (Wang et al., 2025c), while SVD-LLM truncates weight matrices with attention to their output reconstruction (Wang et al., 2025b). Our analysis separates the dimension of a learned correction from the local learning freedom available around the same function. We follow the learned visual-token correction from its formation through matched-function learning and task-level preservation. The relationship to these lines of work is developed in Appendix A.

Our main findings are as follows.

- Learned hidden–readout alignment makes a wide adapter act predominantly along one output direction.
- Compact representability and learning freedom differ, as shown by one-unit training and readouts compared at exactly matched functions.
- Retained correction energy alone does not determine whether a low-dimensional projection preserves NLL and individual predictions.

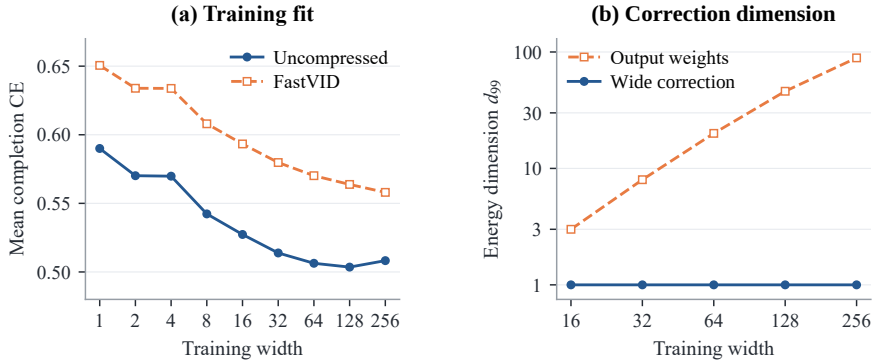

Figure 2. **Width improves fitting while corrections remain narrow.** (a) Mean completion CE over 4,000 updates. (b) Output-weight and correction d_{99} without compression.

2 Experimental Setup

2.1 Hidden computation and readout structure

For a visual token $z \in \mathbb{R}^d$ from the projector, let $\bar{z}$ denote its root-mean-square (RMS) normalization. The residual adapter computes

$$h_\theta(z) = \mathrm{SiLU}(W_d \bar{z} + b_d), \qquad \delta_\theta(z) = W_u h_\theta(z) + b_u, \qquad z' = z + \delta_\theta(z). \tag{1}$$

We call $\delta_\theta(z)$ the *correction* and the affine map from $h_\theta(z)$ to $\delta_\theta(z)$ the *readout*. Hidden width r counts the units in h_θ. Wide uses an unrestricted readout with $r = 64$ unless stated otherwise. Width_1 uses the same architecture with one hidden unit. Base omits the adapter. Task-trained adapters are learned from initialization, while constructed Width_1 instances are obtained from Wide without further task-loss optimization.

The adapter sits at the projector–decoder boundary. This makes its correction explicit and allows projection, removal, and reimplementation while the backbone remains frozen. Within each comparison, we hold the backbone, projector, input condition, and visual-token budget fixed.

Direct retains the hidden width of Wide but restricts the complete correction, including bias, to a single learned axis,

$$\delta_{\mathrm{Direct}}(z) = u\,[v^\top h_\phi(z) + \beta], \qquad \|u\|_2 = 1. \tag{2}$$

The axis is shared across tokens, while the coefficient depends on each input token. Direct task training updates the hidden features, scalar head, and axis. At $r = 64$, Wide has $129d + 64$ adapter parameters and Direct has $65d + 129$. Unlike Direct, Width_1 has an unrestricted output bias, so its raw correction can have rank two. Hidden width and correction dimension are distinct.

2.2 Correction dimension and projection

For each trained Wide checkpoint, we fit orthonormal correction directions U_k on a fixed label-free video set and evaluate

$$\delta_{\mathrm{Keep}_k}(z) = U_k U_k^\top \delta_\theta(z), \qquad \delta_{\mathrm{Remove}_k}(z) = (I - U_k U_k^\top)\delta_\theta(z). \tag{3}$$

Keep_k retains at most k output directions, while Remove_k removes them. Both leave the hidden computation unchanged. Wide, Width_1, and Direct denote architectures. Keep and Remove denote interventions.

We summarize energy concentration by d_{99}, the minimum number of singular directions carrying 99% of squared singular-value energy, and by the leading energy share E_1. These are energy measures rather than algebraic rank. Raw spectra use uncentered corrections, while globally centered spectra subtract the mean correction. Output-weight spectra use W_u. The alignment analysis also measures centered hidden features and corrections on the gradient-evaluation set. We assess preservation through completion NLL, generated-answer accuracy, and prediction agreement.

2.3 MODELS, DATA, AND EVALUATION

Our primary model is Qwen2.5-VL-7B (Bai et al., 2025). We also evaluate Qwen2.5-VL-3B and InternVL2.5-2B (Chen et al., 2024). We use uncompressed inputs and four compression methods. VisionZip (Yang et al., 2025b), FastVID (Shen et al., 2026), and UniComp (Yuan et al., 2026) retain 10% of the visual tokens before the language model. FlashVID (Fan et al., 2026) uses a 10% average token budget, with 12.5% entering the language model before further pruning inside it. All conditions use 32 uniformly sampled frames.

Training uses the same ordered set of 4,000 video-question examples from LLaVA-Video, Video-Marathon, and LongViTU (Zhang et al., 2024; Lin et al., 2026; Wu et al., 2025). Each run uses 4,000 updates with one example per update and completion cross-entropy (CE). Paired comparisons share the data order and schedule shape, while Wide and Direct use the same hidden initialization and a peak learning rate of 3×10^{-4}. Training rates and the one-unit sweep are specified in Appendices B.2 and C.4.

We report training completion CE averaged over all 4,000 updates and development completion NLL on a fixed 512-video set. Generated-answer accuracy is evaluated on MVBench, Video-MME, MLVU, and LongVideoBench (Li et al., 2024a; Fu et al., 2025; Zhou et al., 2025; Wu et al., 2024a). Data splits, initialization, scoring, and uncertainty calculations are specified in Appendix B.

3 TRAIN WIDE, ACT NARROW

3.1 WIDTH IMPROVES FITTING WHILE CORRECTIONS REMAIN NARROW

Increasing hidden width lowers mean training completion CE under both uncompressed and FastVID inputs, most consistently under FastVID, as shown in Figure 2(a). Output-weight dimension grows with width, whereas the correction has $d_{99} = 1$ at every plotted uncompressed endpoint in Figure 2(b). At the default Wide endpoint, output weights have $d_{99} = 20$ and centered hidden features have $d_{99} = 26$. The computation expands without a comparable increase in correction dimension.

The dominant direction carries useful adaptation. Across four compression methods, Wide improves mean accuracy over the four benchmarks by 1.6 to 2.7 percentage points. Keeping only the dominant direction preserves nearly all of this gain in Table 1. Removing it instead returns performance close to Base in Table C.1.

The correction also remains concentrated in Qwen2.5-VL-3B and InternVL2.5-2B, with model-family results in Appendix C.2. In a separate natural-language evaluation, one-direction output-weight projection preserves the reduction in completion NLL, while removing that direction returns NLL to Base in Appendix H.3.

3.2 CONCENTRATION THROUGH HIDDEN–READOUT ALIGNMENT

Across six Wide endpoints under FastVID and UniComp, centered hidden features have d_{99} between 13 and 30, while output weights have d_{99} between 33 and 47. Their correction is much narrower. Let p be the share of hidden variance in the dominant hidden mode, and let κ compare its readout gain with the variance-weighted mean gain of the remaining modes. The share of centered correction energy attributable to this mode is

$$\rho = \frac{p\kappa}{p\kappa + 1 - p}. \tag{4}$$

If $\kappa = 1$, the mode contributes only its hidden-variance share p. At the learned endpoints, the readout favors this mode by a factor of about 9 to 20, raising its share of centered correction energy to 97.1 to 99.6%. The distinction between this attribution and the measured leading correction energy is detailed in Appendix H.5.

This preference forms during training. In a representative FastVID run, κ rises from 0.59 after the first update to 18.47 by 500 updates in Figure 3(a). Across all six runs, raw correction d_{99} falls from 30–41 to 1 or 2 over the same interval.

Table 1. **Keeping the dominant direction preserves average benchmark accuracy.** Qwen2.5-VL-7B generated-answer accuracy (%) under the baseline training configuration. Avg. Score averages the four benchmarks. Avg. Rel. averages their accuracies relative to uncompressed Base.

Visual condition	Execution	MVBench	Video-MME	MLVU	LongVideo Bench	Avg. Score	Avg. Rel. (%)
Uncompressed	Base	61.23	55.81	56.21	52.80	56.51	100.0
	Wide	61.98	56.30	59.15	50.64	57.02	**100.8**
	Keep_1	61.83	56.81	59.66	50.41	57.18	**101.1**
VisionZip	Base	51.75	47.30	50.18	46.00	48.81	86.4
	Wide	53.15	50.37	56.16	46.15	51.46	**91.1**
	Keep_1	53.35	50.41	56.03	45.92	51.43	**91.0**
FastVID	Base	51.51	47.93	51.10	46.22	49.19	87.1
	Wide	53.20	51.26	55.93	46.90	51.82	**91.8**
	Keep_1	52.95	51.15	56.26	47.27	51.91	**91.9**
UniComp	Base	51.26	47.15	50.74	46.60	48.94	86.7
	Wide	51.58	50.26	56.76	46.82	51.35	**91.0**
	Keep_1	51.63	50.30	57.04	46.60	51.39	**91.0**
FlashVID	Base	59.08	51.85	54.42	51.31	54.16	95.8
	Wide	59.98	54.67	58.88	49.59	55.78	**98.6**
	Keep_1	59.70	54.70	58.74	49.44	55.65	**98.4**

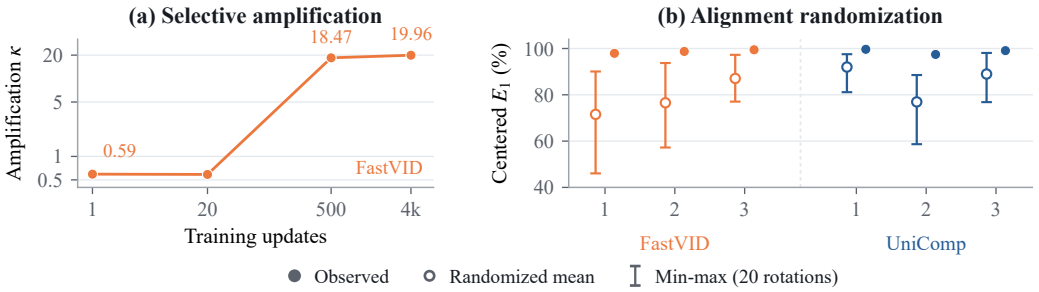

Figure 3. **Learned hidden–readout alignment concentrates the correction.** (a) Relative amplification during training. (b) Centered E_1 before and after readout rotations. Whiskers show the range over 20 rotations, not confidence intervals.

To test whether the factor spectra suffice, we rotate the readout relative to the hidden features while preserving both spectra. All 20 randomized alignments produce lower centered E_1 than the learned pairing at every endpoint in Figure 3(b). The extra concentration comes from how the factors align.

Concentration also forms from broad nonzero Gaussian output weights. Training reduces raw correction d_{99} from 60 to 3 or 4 while improving accuracy and NLL. The learned change from initialization has nearly the same concentrated spectrum as the final correction. This control and the architecture checks are reported in Appendices H.4 and H.2.

3.3 A STRONG ONE-UNIT CORRECTION THAT TASK TRAINING FAILS TO FIND

The dominant hidden mode provides a way to construct a smaller adapter. We approximate its scalar coefficient by an affine function of RMS-normalized visual tokens and implement that approximation in the original one-hidden-unit SiLU architecture. No further task-loss optimization is used. Across three Wide runs, the constructed Width_1 adapters retain about 90% of the NLL improvement over Base, with the construction specified in Appendix H.6.

The constructed adapter computes its correction with that single SiLU unit, without running Wide at inference. Projecting the output of Wide would still require its wide hidden computation and could

not establish what the smaller architecture itself can express. This construction makes the comparison with task-trained Width_1 a test within the same architecture.

Task training of the same architecture does not find this correction under the tested settings. Across four rates, its best development NLL is 1.403, compared with 1.200 for constructed Width_1 without further task-loss optimization. Task-trained MVBench accuracy remains near Base throughout the sweep. The complete results and reference values appear in Appendix C.4.

4 Same Function, Different Learning Spaces

4.1 Direct learns useful one-axis solutions

Direct retains wide hidden computation and learns useful one-axis corrections from initialization, unlike task-trained Width_1. Wide and Direct both improve on Base on Video-MME and MLVU under FastVID and UniComp. Their generated-answer accuracies are similar at the shared learning rate in Table 2. Narrowing the hidden layer and constraining the readout are therefore not interchangeable reductions.

Direct does not reproduce the predictions of the corresponding Keep_1 projection, but independent Wide runs also disagree in Appendix E.2. We therefore hold the correction and hidden features exactly fixed to isolate the effect of readout structure.

4.2 One function, two readouts

Direct and the unrestricted Wide lift implement the shared correction

$$\delta(z) = u\, a(z), \qquad a(z) = v^\top h(z) + \beta, \qquad \|u\|_2 = 1. \tag{5}$$

The coefficient $a(z)$ sets the signed correction for each token. Coefficients across tokens form a *coefficient pattern*. Direct uses (u, v, β, h). The Wide lift uses output weight $uv^\top$, bias $u\beta$, and the same hidden features. A projected Wide checkpoint gives $v = W_u^\top u$ and $\beta = u^\top b_u$. At a directly trained endpoint, these are the learned Direct parameters. Both constructions preserve the function exactly.

We call changes orthogonal to the current output axis u *transverse*. Under Direct, the first-order change is

$$\dot\delta(z) = u\, \dot a(z) + \dot u\, a(z), \qquad u^\top \dot u = 0. \tag{6}$$

The first term changes coefficients along u. The second rotates the axis through the existing coefficient pattern. The resulting distinction between Direct and the Wide lift is illustrated in Figure 1.

4.3 Different local learning spaces

Collect hidden features over n token observations and append a constant row for the output bias, giving $\widetilde H \in \mathbb{R}^{(r+1)\times n}$ with rank q. Its row space contains the supported coefficient patterns. Thus q counts independent token patterns, not output directions. With $C = [W_u\ b_u]$, the unrestricted readout gives $D = C\widetilde H$. At the shared function, $D = ua^\top$ with $a \neq 0$ and $a^\top \in \mathrm{row}(\widetilde H)$. The local learning space consists of first-order changes in D obtainable by varying the readout with $\widetilde H$ fixed.

For two tokens with equal current coefficients but different hidden features, Direct rotates their corrections together. The Wide lift can move them differently along a new direction. The proposition makes this restriction precise.

Proposition 1. Local learning spaces at a shared correction. The unrestricted and one-axis readouts have local learning spaces

$$\begin{aligned} \mathcal{T}_{\mathrm{W}} &= \{X \mid \mathrm{row}(X) \subseteq \mathrm{row}(\widetilde H)\}, \\ \mathcal{T}_1 &= \{us^\top + ba^\top \mid s^\top \in \mathrm{row}(\widetilde H),\ u^\top b = 0\}. \end{aligned} \tag{7}$$

Their dimensions are dq and $q + d - 1$. The inclusion $\mathcal{T}_1 \subseteq \mathcal{T}_{\mathrm{W}}$ is strict when $d > 1$ and $q > 1$.

Table 2. **Wide and Direct learn useful corrections at the shared learning rate.** Generated-answer accuracy (%). Adapted models report mean ± sample standard deviation over three runs.

	FastVID			UniComp		
Benchmark	Base	Wide	Direct	Base	Wide	Direct
MVBench	51.51	52.45 ± 0.49	53.16 ± 0.31	51.26	50.66 ± 0.96	51.21 ± 0.34
Video-MME	47.93	50.28 ± 0.95	50.21 ± 0.12	47.15	50.12 ± 0.28	49.95 ± 0.08
MLVU	51.10	57.10 ± 0.78	56.78 ± 0.71	50.74	55.95 ± 1.23	56.00 ± 0.85
LongVideoBench	46.22	45.25 ± 0.33	45.45 ± 0.69	46.60	46.95 ± 0.86	46.67 ± 0.45

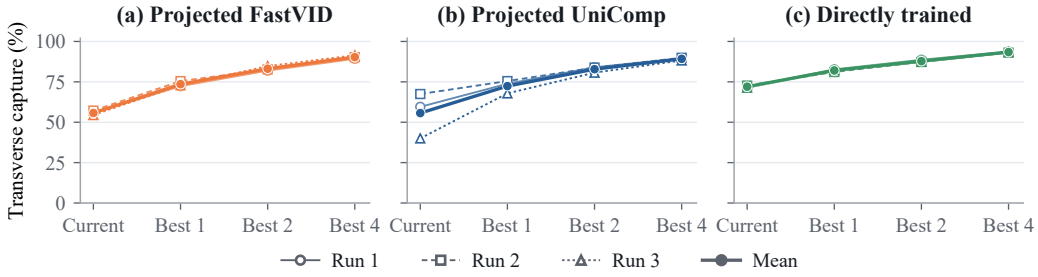

Figure 4. **One coefficient pattern misses loss-relevant transverse changes.** Current uses the fitted pattern and Best k uses k optimal patterns. Fractions use wide-accessible transverse energy. Thin curves show three runs and thick curves their mean.

Both readouts can change coefficients along u. Direct can move transversely only through the current pattern a, while the Wide lift can use any pattern supported by the hidden features. Updating the hidden features in Direct changes $a(z)$ and contributes along u at first order, leaving the transverse restriction intact. The proof and rank-k extension are in Appendix I.

4.4 Gradients use additional coefficient patterns

A larger local learning space matters only if the loss uses it. At the shared function, let $G = \nabla_D \mathcal{L}$, let P_H project onto $\mathrm{row}(\widetilde{H})$, and define $P_a = aa^\top/\|a\|_2^2$ and $P_\perp = I - uu^\top$. Frobenius-orthogonal projection onto the two readout spaces gives

$$\Pi_{\mathrm{W}} G = GP_H, \qquad \Pi_1 G = uu^\top GP_H + P_\perp GP_a. \tag{8}$$

The two readouts have the same current function and correction-space gradient G. The projections identify the loss-relevant changes available to each readout. Their squared norms measure first-order descent in the correction-space Frobenius metric, not the parameter update produced by AdamW.

Let $M = P_\perp GP_H$ be the *wide-accessible transverse gradient*. For a unit coefficient pattern s in the feature row space, its captured fraction and the best single-pattern capture are

$$c(s) = \frac{\|Ms\|_2^2}{\|M\|_F^2}, \qquad c_1^\star = \frac{\sigma_1(M)^2}{\|M\|_F^2}. \tag{9}$$

The current pattern is $s = a/\|a\|_2$. The best single pattern is chosen after observing the gradient and can replace it entirely. All capture fractions use $\|M\|_F^2$, not total gradient energy, as their denominator.

The restriction appears at directly trained Direct endpoints. Across three such functions, even the best single pattern chosen for the measured gradient leaves 17.3 to 18.9% of wide-accessible transverse gradient energy uncaptured. The six projected Wide functions leave 24.5 to 32.1% uncaptured under the same test. Four patterns capture about 90% across all nine functions in Figure 4. The trained Direct result establishes that the gap does not depend on projecting a Wide checkpoint.

The gradient comparison isolates a loss-relevant advantage of the unrestricted readout at the same function. It measures first-order changes while the hidden features are fixed. Matched continuation then tests whether that extra freedom improves fitting under the shared training schedule. The Width_1 comparison concerns a different constraint because it also narrows the hidden computation.

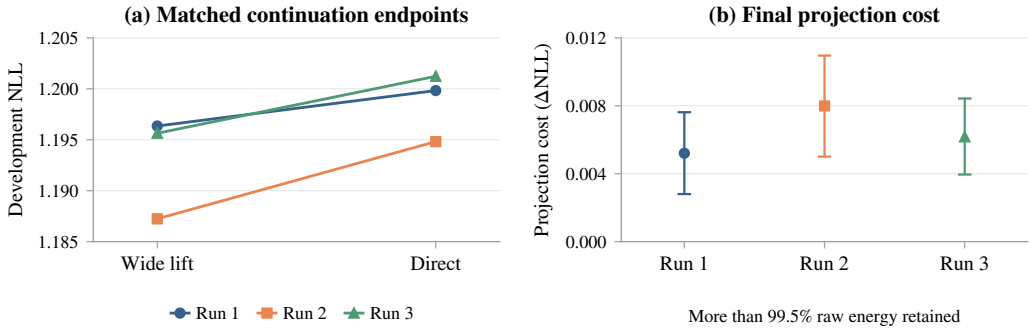

Figure 5. **Small correction components contribute to likelihood.** (a) Paired 500-update NLL. (b) Final dominant-direction projection cost with paired 97.5% intervals.

The same restriction persists along the baseline training trajectories even when raw and centered Wide corrections already have $d_{99} = 1$. Here the current one-axis space excludes about 38% of total wide-accessible gradient energy without compression and 52% under FastVID. This trajectory measure uses the current pattern and includes along-axis energy in its denominator, unlike the optimal transverse capture above. Its definition and full trajectories are given in Appendix F.4.

5 Training and Preserving Compact Corrections

5.1 Small components contribute to the continuation advantage

We initialize Direct and the Wide lift from the Keep_1 projection of each of three shared-rate FastVID Wide checkpoints, with hidden features fixed. We update only the unrestricted readout or the Direct axis and scalar head. With matched data and optimization settings, the Wide lift reaches lower development NLL than Direct after 500 updates in all three runs, as shown in Figure 5(a). Relative to the shared start, paired intervals support a Wide-lift improvement in one run and a Direct deterioration in another. Full trajectories and contrasts appear in Appendices F.5 and F.6.

Each final Wide lift retains more than 99.5% of raw correction energy in its dominant direction. Yet projecting onto that direction raises NLL in every run and removes the point-estimate advantage over Direct in Figure 5(b). Small components of the correction carry part of the benefit gained by the unrestricted readout. Because this projection minimizes correction reconstruction error rather than task NLL, it tests whether the dominant direction preserves the fitted Wide-lift behavior. Final spectra and contrasts appear in Appendix F.6.

5.2 Energy retention does not ensure behavioral preservation

The leading direction of the original Wide endpoints carries 97.4 to 99.1% of raw correction energy, yet retaining it alone raises development NLL by 0.017 to 0.024. The cost falls as more directions are kept in Figure 6. At one checkpoint, generated-choice agreement with Wide rises from 92.4 to 96.3% when four directions are retained, while MVBench accuracy changes little in Table C.5. A direction can carry nearly all correction energy without preserving the fitted predictions.

Paired 95% intervals exclude zero for all three one-direction and two-direction costs. At four directions, only one of three intervals remains above zero. Even when $d_{99} = 1$, retaining the leading direction alone can raise development NLL.

The token coefficients matter as well as the output direction. They determine the strength and sign of the correction at each token. At the decoder-analysis checkpoint, replacing them with a per-video constant of equal RMS keeps option accuracy near 52%, but lowers option agreement with Wide from 97.8 to 85.2% in Table G.2. Similar option accuracy can conceal different predictions even when the direction and per-video RMS are retained.

These effects extend beyond the default initialization and model family. Starting from broad random output weights also produces concentrated corrections whose one-direction projections raise NLL and change generated choices in Appendix H.4. In InternVL2.5-2B, three directions improve generated-

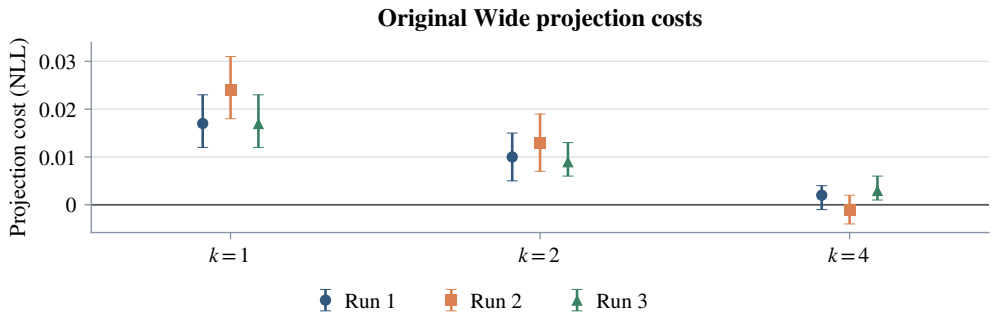

Figure 6. **More directions better preserve the original Wide endpoints.** NLL cost of retaining k correction directions across three FastVID runs, with paired 95% intervals.

answer accuracy by 0.63 percentage points over one direction even though the leading direction carries 99.5% of raw energy in Appendix H.1. The retained dimension should be checked against likelihood and predictions, since correction energy alone does not determine which small components affect behavior.

6 Discussion

The wide adapters studied here learn corrections concentrated almost entirely along one output direction. Hidden–readout alignment explains how the concentrated output forms, while the constructed one-unit adapter shows that much of it can be implemented compactly. Its weaker task-trained counterpart shows that representability alone does not settle how easily the correction is learned. At the same function, the unrestricted readout offers loss-relevant changes that the one-axis readout cannot make through its current coefficient pattern. Direct still learns useful one-axis solutions, so hidden width and readout freedom must be considered separately.

This distinction changes two practical decisions. Adapter width should be assessed by what training can reach, not inferred from the number of dominant output directions after training. The number of directions retained should be assessed by completion NLL and predictions as well as correction energy. In our frozen video-language setting, one direction can preserve average accuracy while missing small components that matter for likelihood or individual answers. The one-unit construction removes wide hidden computation, while a one-direction projection retains it. The spectrum can suggest candidate retained ranks, and comparisons with Wide on paired development NLL and individual predictions reveal whether each compact correction preserves the fitted behavior. Future work can examine how training selects hidden–readout alignment and how the decoder responds to those small components. A simple final correction can require a richer path to learn and more than its dominant direction to preserve.

AI USE STATEMENT

We used generative AI tools to assist with editing, polishing, and shortening text written by the authors, as well as with executing code during experiments. The research question, methodology, and final experimental design were determined and reviewed by the authors. No experimental results were generated, modified, or selected by an AI tool, and all reported numbers were taken directly from the evaluation logs. The authors reviewed all AI-assisted work. AI-assisted edits were checked for consistency with the experiments, and the reported results were verified against the underlying data and evaluation logs. The authors take full responsibility for the final content of this work, including any text or other materials produced with the assistance of generative AI.

ETHICS STATEMENT

This work does not raise any specific ethical concerns beyond the standard considerations associated with existing pretrained models and benchmark datasets.

REPRODUCIBILITY STATEMENT

The main text specifies the experimental setup and primary comparisons. Appendix B documents the data splits, adapter initialization, training and evaluation procedures, and uncertainty calculations. Appendices C–H report additional numerical results, controls, construction details, and supporting analyses. Appendix I contains the proofs and derivations for the local learning-space and gradient results.

REFERENCES

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pp. 7319–7328, 2021.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International conference on machine learning*, pp. 244–253. PMLR, 2018.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in neural information processing systems*, 32, 2019.

Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-VL technical report. arXiv:2502.13923, 2025. URL `https://arxiv.org/abs/2502.13923`.

Jinhe Bi, Yujun Wang, Haokun Chen, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. Llava steering: Visual instruction tuning with 500x fewer parameters through modality linear representation-steering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15230–15250, 2025.

Yuxiao Chen, Jue Wang, Zhikang Zhang, Jingru Yi, Xu Zhang, Yang Zou, Zhaowei Cai, Jianbo Yuan, Xinyu Li, Hao Yang, et al. Learning compact video representations for efficient long-form video understanding in large multimodal models. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4242–4252. IEEE, 2026.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

Ziyang Fan, Keyu Chen, Ruilong Xing, Yulin Li, Li Jiang, and Zhuotao Tian. Flashvid: Efficient video large language models via training-free tree-based spatiotemporal token merging. *arXiv preprint arXiv:2602.08024*, 2026.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

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

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22195–22206, 2024a.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European conference on computer vision*, pp. 323–340. Springer, 2024b.

Zhiyuan Li, Yuping Luo, and Kaifeng Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. *arXiv preprint arXiv:2012.09839*, 2020a.

Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joey Gonzalez. Train big, then compress: Rethinking model size for efficient training and inference of transformers. In *International Conference on machine learning*, pp. 5958–5968. PMLR, 2020b.

Jingyang Lin, Jialian Wu, Ximeng Sun, Ze Wang, Jiang Liu, Yusheng Su, Xiaodong Yu, Hao Chen, Jiebo Luo, Zicheng Liu, et al. Unleashing hour-scale video training for long video-language understanding. *Advances in Neural Information Processing Systems*, 38:17523–17552, 2026.

Zhihang Liu, Chen-Wei Xie, Pandeng Li, Liming Zhao, Longxiang Tang, Yun Zheng, Chuanbin Liu, and Hongtao Xie. Hybrid-level instruction injection for video token compression in multi-modal large language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8568–8578. IEEE, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, et al. Fastvid: Dynamic density pruning for fast video large language models. *Advances in Neural Information Processing Systems*, 38:123553–123581, 2026.

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. Lora vs full fine-tuning: An illusion of equivalence. *Advances in Neural Information Processing Systems*, 38: 174627–174662, 2026.

Bart Vandereycken. Low-rank matrix completion by riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

Han Wang, Yuxiang Nie, Yongjie Ye, Yanjie Wang, Shuai Li, Haiyang Yu, Jinghui Lu, and Can Huang. Dynamic-vlm: Simple dynamic visual token compression for videollm. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20812–20823. IEEE, 2025a.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. Svd-llm: Truncation-aware singular value decomposition for large language model compression. In *International Conference on Learning Representations*, volume 2025, pp. 19299–19319, 2025b.

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Lora-pro: Are low-rank adapters properly optimized? In *International Conference on Learning Representations*, volume 2025, pp. 93787–93808, 2025c.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024a.

Rujie Wu, Xiaojian Ma, Hai Ci, Yue Fan, Yuxuan Wang, Haozhe Zhao, Qing Li, and Yizhou Wang. Longvitu: Instruction tuning for long-form video understanding. *arXiv preprint arXiv:2501.05037*, 2025.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. *Advances in Neural Information Processing Systems*, 37:63908–63962, 2024b.

Chenyu Yang, Xuan Dong, Xizhou Zhu, Weijie Su, Jiahao Wang, Hao Tian, Zhe Chen, Wenhai Wang, Lewei Lu, and Jifeng Dai. Pvc: Progressive visual token compression for unified image and video processing in large vision-language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24939–24949. IEEE, 2025a.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802. IEEE, 2025b.

Chao Yuan, Shimin Chen, Minliang Lin, Limeng Qiao, Guanglu Wan, and Lin Ma. Unicomp: Rethinking video compression through informational uniqueness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18609–18618, 2026.

Shaolei Zhang, Qingkai Fang, Yang Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token. In *International Conference on Learning Representations*, volume 2025, pp. 53285–53310, 2025.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701. IEEE, 2025.

A Related Work

Visual token compression and learned interfaces. Visual token compression reduces inference cost but can remove useful information. Existing methods use token selection, merging, or learned compact representations (Zhang et al., 2025; Li et al., 2024b). Learned compression modules and training on compressed inputs address the efficiency–accuracy tradeoff (Chen et al., 2026; Wang et al., 2025a; Yang et al., 2025a; Liu et al., 2025). We examine the correction learned by a residual adapter across uncompressed inputs and multiple compression methods. These conditions help distinguish its learning geometry from any particular compression mechanism.

Parameter-efficient adaptation. Low-rank weight updates, bottleneck adapters, and intrinsic-dimension methods examine how much trainable freedom is sufficient for adaptation (Hu et al., 2021; Houlsby et al., 2019; Aghajanyan et al., 2021). Analyses of low-rank and full fine-tuning also distinguish nominal rank, effective update dimension, and functional differences (Shuttleworth et al., 2026). Here the measured object is the realized correction in representation space. The same one-unit architecture supports a strong constructed correction and a weaker task-trained solution, separating representation from learning.

Low-dimensional representation control. Representation finetuning learns interventions on a frozen model, including interventions restricted to a low-rank subspace (Wu et al., 2024b). Visual representation steering likewise trains compact interventions for multimodal instruction tuning (Bi et al., 2025). Other work identifies a single activation direction that mediates a particular behavior (Arditi et al., 2024). We analyze a narrow correction that emerges from an unrestricted readout, then compare learning at the same function and preservation after projection.

Optimization under factorized parameterizations. Low-rank adaptation can be analyzed through its equivalent gradient in full weight space. Adjusting the two factor gradients can bring it closer to the full fine-tuning gradient (Wang et al., 2025c). We fix the hidden features and current correction and measure the loss-relevant changes available to each readout in correction space. The shared-function construction and directly trained Direct endpoints separate this comparison from differences between independently fitted solutions.

Overparameterization and post-training compression. Overparameterization can improve optimization, and gradient descent can favor low-rank solutions in matrix factorization (Arora et al., 2018; 2019; Li et al., 2020a). Training large models before compression can combine faster optimization with efficient inference (Li et al., 2020b). Activation-aware singular-value truncation uses calibration inputs to control layer-output reconstruction error (Wang et al., 2025b). We project the realized correction and compare its retained energy with preservation of completion likelihood and individual predictions.

B Data, Training, and Evaluation

B.1 Training data and analysis sets

Unless stated otherwise, task-trained adapters use 4,000 optimizer updates on the same ordered set of 4,000 video-question pairs from 4,000 unique videos. Initialization seeds are 43, 45, and 46, with seed 43 used for the baseline breadth experiments. Multiple-choice targets use the correct option letter, while natural-language targets retain the source answer. All models receive 32 uniformly sampled video frames. We remove byte-identical overlaps between the training data and evaluation media before training. The training data composition is summarized in Table B.1.

The held-out sets used throughout the analysis are summarized in Table B.2. Correction directions are fitted without answer labels and held fixed during evaluation, with each checkpoint using its own fitted correction basis. Natural-language experiments instead use the output-weight basis described in Appendix H.3.

Table B.1. **Training data composition.**

Source	Choice	Free-form	Total
LLaVA-Video	1,474	1,127	2,601
VideoMarathon	1,196	3	1,199
LongViTU	0	200	200
Total	2,670	1,330	4,000

Table B.2. **Evaluation and analysis sets.** Except for the nested 200-example generation subset, the listed sets use disjoint videos.

Set	Size	Primary use
Development	512 videos	Completion NLL and rate selection
Correction basis	160 videos	Correction directions
Gradient evaluation	400 videos	Local learning space
Natural-language evaluation	1,000 examples	Completion NLL and generation on 200
Decoder discovery	400 videos	Direction and coefficient analysis
Decoder confirmation	512 videos	Held-out confirmation

B.2 Adapter initialization and training

For a visual token $z \in \mathbb{R}^d$ from the projector, the normalization and adapter dimensions are

$$\bar{z} = \frac{z}{\sqrt{\|z\|_2^2/d + \epsilon}}, \qquad W_d \in \mathbb{R}^{r\times d}, \qquad W_u \in \mathbb{R}^{d\times r}. \tag{B.1}$$

Wide is initialized with $W_u = b_u = 0$, random hidden weights, and a nonzero random hidden bias. Direct is initialized with a unit-norm learned axis and a zero scalar readout and bias. The axis is normalized on every forward pass. Matched Wide and Direct runs share the same initial hidden weights and biases, and both begin from zero correction. Each run starts with fresh optimizer and scheduler states.

Width_1 has one hidden unit, whereas Direct retains hidden width 64. Because Width_1 uses an unrestricted affine readout, its correction has the form $w\, s(z) + b_u$. The raw correction can therefore have rank two. Centering removes the shared bias and leaves rank at most one. Hidden width and correction dimension are therefore treated separately.

Task-trained and constructed Width_1 instances use the same one-hidden-unit SiLU architecture. The constructed instances are obtained from fitted Wide adapters without further task-loss optimization, as detailed in Appendix H.6.

For video v_i, prompt P_i, serialized completion s_i, and supervised token positions T_i, the training objective is

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|T_i|}\sum_{t\in T_i} -\log p_\theta(s_{it} \mid v_i, P_i, s_{i,<t}). \tag{B.2}$$

Loss is computed only over the assistant completion, with prompt and visual-token positions masked. It is averaged within each example and then across examples. Qwen and InternVL use their native chat templates.

All reported learning rates are peak rates. AdamW (Loshchilov & Hutter, 2017) uses betas $(0.9, 0.999)$, epsilon 10^{-8}, zero weight decay, and global gradient-norm clipping at 1.0. The backbone uses BF16 and the adapter uses FP32. A standard 4,000-update run starts at 1% of the peak rate, warms up linearly for 100 updates, and then follows cosine decay over the remaining 3,900 updates, ending at 10% of the peak rate. The training configurations used across the experiments are summarized in Table B.3.

The Width_1 learning-rate comparison uses four peak rates from 10^{-5} to 3×10^{-4} under the same 4,000-update schedule, as reported in Table C.9. The 500-update continuation experiments use the shared peak rate and are described in Appendix F.5.

Table B.3. **Training configurations and their primary uses.**

Configuration	Peak LR	Updates	Primary use
Baseline	10^{-5}	4,000	Width, breadth, and training-time gradients
Decoder analysis	3×10^{-5}	4,000	Decoder interventions
Shared rate	3×10^{-4}	4,000	Replicated Wide and Direct runs
Continuation	3×10^{-4}	500	Fixed-feature continuation of Wide lift and Direct

B.3 Evaluation

Development completion NLL is computed on the fixed 512-video development set with 9,501 supervised tokens using full-vocabulary cross-entropy without option renormalization. Each example receives equal weight. Generated-answer accuracy compares parsed greedy responses with the gold answer. Generated-choice agreement compares valid parsed choices between models. Option accuracy and option agreement instead use the first-position option argmax.

B.4 Uncertainty and spectral measures

Unless stated otherwise, reported intervals are 95% video-cluster bootstrap intervals from 10,000 draws. All questions and interventions from a sampled video are retained as a unit. Three-run task contrasts average the fitted models within each draw, while sample standard deviation separately summarizes variation across initialization seeds. Multiplicity-adjusted comparisons state the comparison family and interval level locally.

For singular values σ_j of a specified matrix, we define

$$E_k = \frac{\sum_{j \le k} \sigma_j^2}{\sum_j \sigma_j^2}, \qquad d_{99} = \min\{k \mid E_k \ge 0.99\}. \tag{B.3}$$

C Benchmark Breadth and Compact Execution

C.1 Baseline gains and removal

The compressed-input gains reported in Table 1 persist after projection onto one output direction. The corresponding removal results and paired intervals for Wide minus Base are reported in Tables C.1 and C.2. For the model-family comparisons below, average relative accuracy is

$$\text{Avg. Rel.}_m = \frac{1}{4} \sum_b 100 \, \frac{s_{m,b}}{s_{0,b}}, \tag{C.1}$$

where $s_{0,b}$ is the uncompressed Base accuracy on benchmark b.

Table C.1. Generated-answer accuracy after Remove_1 under the baseline configuration (%).

Condition	MVBench	Video-MME	MLVU	LongVideoBench
Uncompressed	61.43	55.93	55.93	52.95
VisionZip	51.55	47.07	50.41	46.07
FastVID	51.60	47.93	51.15	46.37
UniComp	51.25	47.41	50.87	46.52
FlashVID	59.13	51.82	54.23	51.76

Across the compressed conditions, question-weighted gains of Wide over Base range from 1.82 to 2.65 percentage points and remain supported after Holm adjustment.

Table C.2. Paired 95% intervals for Wide minus Base generated-answer accuracy (pp).

Condition	MVBench	Video-MME	MLVU	LongVideoBench
Uncompressed	[−0.40, 1.90]	[−1.00, 1.96]	[1.12, 4.71]	[−4.63, 0.22]
VisionZip	[0.30, 2.50]	[1.67, 4.56]	[4.14, 7.82]	[−2.22, 2.50]
FastVID	[0.45, 2.80]	[1.96, 4.70]	[2.88, 6.76]	[−1.77, 3.05]
UniComp	[−1.05, 1.55]	[1.63, 4.59]	[4.10, 7.90]	[−2.23, 2.66]
FlashVID	[−0.23, 2.05]	[1.33, 4.30]	[2.51, 6.39]	[−3.91, 0.53]

Table C.3. Generated-answer accuracy across model families (%), shown as Base → Wide. Avg. Rel. uses the uncompressed Base of the same model.

Condition	MVBench	Video-MME	MLVU	LongVideoBench	Avg. Rel.(%)
Qwen2.5-VL-3B					
Uncompressed	59.78→59.58	51.70→52.63	55.06→55.70	49.51→48.62	100.0→100.2
FastVID	50.50→50.73	44.30→46.56	48.30→51.47	45.40→45.77	87.4→90.2
UniComp	50.40→50.78	44.37→46.26	48.11→51.84	45.18→45.10	87.2→89.9
InternVL2.5-2B					
Uncompressed	67.70→68.00	52.44→53.11	59.25→58.46	48.24→48.24	100.0→100.1
FastVID	47.70→50.50	44.19→44.96	52.71→54.14	44.95→45.70	84.2→86.6
UniComp	59.13→60.28	44.48→45.07	52.90→52.99	43.68→43.38	88.0→88.6

C.2 Model scale and decoder family

The baseline evaluation extends to Qwen2.5-VL-3B and InternVL2.5-2B, as reported in Table C.3. Qwen2.5-VL-3B also shows lower option NLL for Wide than Width_1 without compression and under both FastVID and UniComp. Under FastVID in InternVL2.5-2B, Wide improves the four-benchmark mean over Width_1 by 1.40 percentage points [0.684, 2.107]. Additional retained-direction results for InternVL2.5-2B are reported in Appendix H.1.

C.3 Shared-rate spectra and retained directions

Table C.4. Shared-rate correction spectra on the 160 basis-fitting videos. Energy shares are percentages.

Condition	Seed	Raw E_1	Raw d_{99}	Centered E_1	Centered d_{99}
FastVID	43	97.35	2	97.85	4
	45	98.46	2	98.68	2
	46	99.13	1	99.46	1
UniComp	43	99.53	1	99.66	1
	45	96.93	3	97.56	3
	46	98.83	2	99.11	1

The shared-rate corrections remain concentrated across all six Wide checkpoints reported in Table C.4. The hidden features and readout weights remain substantially higher-dimensional than the correction at the three FastVID checkpoints. Output-weight d_{99} ranges from 45 to 47 and centered hidden-feature d_{99} from 16 to 30, while raw correction d_{99} is only 1 or 2. Retained-direction results for one Wide checkpoint per compressor are reported in Table C.5. Additional directions increase generated-choice agreement with Wide even when the raw correction already has $d_{99} = 1$, despite similar generated-answer accuracy.

Table C.5. **MVBench accuracy and fidelity to Wide at the shared learning rate.** Seed 43. Accuracy and generated-choice agreement are percentages. F_ℓ is centered logit-change cosine similarity.

Condition	Execution	Acc.	Generated-choice agreement	F_ℓ
FastVID	Wide	52.87	100.00	1.000
	Keep_1	52.62	92.44	0.968
	Keep_2	52.74	95.29	0.987
	Keep_4	52.94	96.35	0.992
UniComp	Wide	50.16	100.00	1.000
	Keep_1	50.44	95.22	0.991
	Keep_2	50.46	97.12	0.997
	Keep_4	50.09	97.32	0.998

Table C.6. **Original Wide projection costs on FastVID development data.** Cost is Keep_k minus Wide, with paired 95% intervals.

Seed	Wide NLL	k	Keep_k NLL	Cost	95% interval
43	1.179	1	1.196	0.017	$[0.012, 0.023]$
		2	1.189	0.010	$[0.005, 0.015]$
		4	1.181	0.002	$[-0.001, 0.004]$
45	1.174	1	1.198	0.024	$[0.018, 0.031]$
		2	1.187	0.013	$[0.007, 0.019]$
		4	1.173	-0.001	$[-0.004, 0.002]$
46	1.179	1	1.196	0.017	$[0.012, 0.023]$
		2	1.189	0.009	$[0.006, 0.013]$
		4	1.182	0.003	$[0.001, 0.006]$

Table C.7. Generated-answer accuracy after removing dominant directions on MVBench (%). Shared-rate seed 43.

Execution	FastVID	UniComp
Base	51.51	51.26
Remove_1	51.86	50.96
Remove_2	52.09	50.79
Remove_4	51.79	50.79

Removal results provide the complementary comparison with Base. Retaining additional directions also reduces the development NLL cost of projection in all three shared-rate FastVID runs, as reported in Table C.6.

C.4 Hidden-width comparisons

The width sweep uses hidden widths 1, 2, 4, 8, 16, 32, 64, 128, and 256. Width_1 reduces the hidden computation to one unit, whereas Keep_1 retains the full hidden computation of Wide and restricts only the output correction.

Table C.8. **Qwen2.5-VL-3B MVBench option NLL.** $\Delta = \mathrm{Wide} - \mathrm{Width}_1$.

Condition	Width_1	Wide	Δ	95% interval
Uncompressed	0.953	0.929	-0.024	$[-0.035, -0.013]$
FastVID	1.149	1.103	-0.046	$[-0.059, -0.033]$
UniComp	1.119	1.090	-0.029	$[-0.041, -0.017]$

Table C.9. **A one-unit adapter can express a stronger correction than task training finds.** Qwen2.5-VL-7B with FastVID. Task-trained runs use 4,000 updates.

Task-trained Width_1		
Peak LR	NLL	Accuracy (%)
10^{-5}	1.458	51.54
3×10^{-5}	1.453	51.59
10^{-4}	1.437	51.74
3×10^{-4}	1.403	51.51

Reference corrections	
Configuration	NLL
Base	1.460
Wide at 10^{-5}	1.247
Wide at 3×10^{-4}	1.179
Constructed Width_1	1.200

The four-rate Width_1 comparison and the matched construction comparison use Qwen2.5-VL-7B with FastVID at seed 43 in Table C.9. The learning-rate sweep uses the same 4,000-update schedule at all four rates. The Base MVBench accuracy is 51.51%. In Qwen2.5-VL-3B, Wide also has lower option NLL than Width_1 under uncompressed, FastVID, and UniComp inputs in Table C.8.

Table C.10. Mean training completion CE over 4,000 updates. Seed 43 and peak rate 10^{-5} throughout.

Hidden width	1	2	4	8	16	32	64	128	256
Uncompressed	0.590	0.570	0.570	0.542	0.527	0.514	0.506	0.504	0.508
FastVID	0.651	0.634	0.634	0.608	0.593	0.580	0.570	0.564	0.558

Mean training completion CE over the 4,000-update width sweep is reported in Table C.10. Under FastVID, average training CE decreases across the full sweep. Without compression, it decreases through width 128 before slightly increasing at width 256. At the shared rate, FastVID Width_1 has no generated-answer accuracy gain over Base despite improving completion NLL.

These comparisons concern Width_1 trained from initialization. Post-training constructions within the same architecture are reported in Appendix H.6.

D Direct Training at the Shared Rate

D.1 Task accuracy across three initializations

Wide and Direct use the shared peak learning rate 3×10^{-4}, the same 4,000-update schedule, and matched hidden initialization and data order. Three-run means and sample standard deviations are reported in Table 2, while paired Direct-minus-Wide accuracy differences are shown in Figure D.1. All eight paired 95% intervals include zero. Prediction-fidelity comparisons between Direct and the corresponding Keep_1 projection are provided in Appendix E.

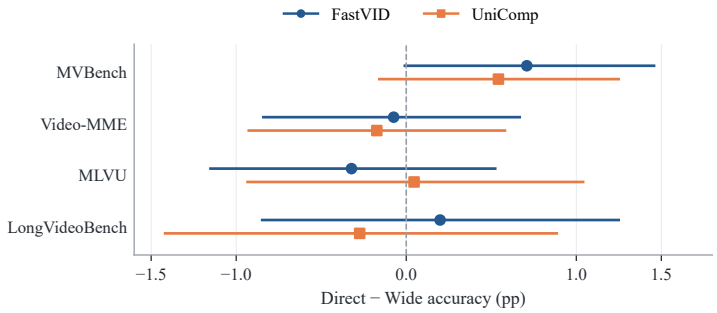

Figure D.1. **Shared-rate accuracy contrasts.** Direct minus Wide generated-answer accuracy across three runs. Error bars show paired 95% intervals.

E Prediction Fidelity Across Runs

E.1 Metrics and shared-rate fidelity

Task accuracy and prediction fidelity measure different aspects of adaptation. We use generated-choice agreement and option agreement as defined in Appendix B, along with centered logit-change cosine similarity, relative error, and flip recall. For Direct, fidelity is measured against the corresponding Keep_1 projection. For Keep_1, the reference is Wide.

For each example, we subtract the Base option logits from those of method X and center this change over options. We concatenate these centered changes into $\Delta\ell_X$ and define

$$F_\ell(X,Y) = \frac{\langle \Delta\ell_X, \Delta\ell_Y \rangle}{\|\Delta\ell_X\|_2 \|\Delta\ell_Y\|_2}, \qquad R_\ell(X,Y) = \frac{\|\Delta\ell_X - \Delta\ell_Y\|_2}{\|\Delta\ell_Y\|_2}. \tag{E.1}$$

The cosine F_ℓ requires both vectors to be nonzero, while R_ℓ requires a nonzero reference $\Delta\ell_Y$. Flip recall is the fraction of reference-induced option changes from Base that the tested method reproduces with the same new choice.

Table E.1. **Shared-rate prediction fidelity on MVBench.** Values are mean ± sample SD over three runs. Agreements and flip recall are percentages.

Comparison	Gen. agree.	Opt. agree.	F_ℓ	R_ℓ	Flip recall
FastVID / Direct–Keep_1	83.64 ± 0.46	82.98 ± 0.81	0.779 ± 0.028	0.646 ± 0.062	60.17 ± 2.75
FastVID / Keep_1–Wide	93.09 ± 0.72	93.85 ± 0.28	0.976 ± 0.006	0.220 ± 0.029	87.19 ± 1.51
UniComp / Direct–Keep_1	80.54 ± 1.83	77.81 ± 2.57	0.694 ± 0.065	0.808 ± 0.198	55.27 ± 8.75
UniComp / Keep_1–Wide	92.78 ± 2.11	92.73 ± 1.83	0.973 ± 0.016	0.223 ± 0.075	85.95 ± 2.72

The shared-rate fidelity comparisons are reported in Table E.1. Across both FastVID and UniComp, Keep_1 agrees more closely with Wide than Direct does with the corresponding Keep_1, with higher generated-choice agreement and centered logit-change cosine similarity and lower relative error.

E.2 Independent retraining as a reference

Table E.2. **Prediction agreement between Keep_1 projections of independently trained Wide runs.** Shared-rate MVBench. Agreements are percentages.

Condition	Seed pair	Gen. choice agree.	Opt. agree.
FastVID	43–45	89.01	85.13
	43–46	86.38	78.67
	45–46	86.76	82.70
UniComp	43–45	83.68	81.68
	43–46	80.45	78.45
	45–46	78.90	78.92

Independent retraining also changes individual predictions. Under FastVID, Keep_1 projections of independently trained Wide runs agree on 86.38 to 89.01% of generated choices, while shared-rate Direct agrees with its corresponding Keep_1 on 83.23 to 84.13%. Under UniComp, agreement across independently trained Keep_1 projections ranges from 78.90 to 83.68%, overlapping the Direct range of 78.55 to 82.15%.

F Completion Gradients and Transverse Pattern Structure

F.1 Shared function and gradient decomposition

The representative endpoint analysis uses the seed-43 Qwen2.5-VL-7B FastVID checkpoint trained at the shared rate 3×10^{-4}. We evaluate the Keep_1 function projected from this checkpoint. The

one-axis and unrestricted readouts implement exactly the same correction and share the same hidden features.

The gradient evaluation uses 400 held-out videos and 16,808 visual-token observations. We hold the post-SiLU features fixed, append the constant bias row, and pool all token observations into $\widetilde{H}$, whose numerical rank is 65. Write the shared correction as $D_1 = ua^\top$, with unit axis u and nonzero coefficient vector a. Let P_H project onto $\mathrm{row}(\widetilde{H})$, $P_a = aa^\top/\|a\|_2^2$, and $P_\perp = I - uu^\top$. For the completion gradient $G = \nabla_{D_1}\mathcal{L}$,

$$\Pi_{\mathrm{W}} G = GP_H, \qquad \Pi_1 G = uu^\top GP_H + P_\perp GP_a. \tag{F.1}$$

The corresponding energy decomposition is

$$A = \|uu^\top GP_H\|_F^2, \qquad B = \|P_\perp GP_a\|_F^2, \qquad C = \|P_\perp GP_H\|_F^2, \tag{F.2}$$

where A is the energy along the current output axis, B is the transverse energy captured through the current coefficient pattern, and C is the full wide-accessible transverse energy. Because $\mathrm{span}(a) \subseteq \mathrm{row}(\widetilde{H})$,

$$E_{\mathrm{W}} = A + C, \qquad E_{\mathrm{Direct}} = A + B, \qquad E_{\mathrm{extra}} = C - B = \|P_\perp G(P_H - P_a)\|_F^2. \tag{F.3}$$

Table F.1. **Gradient-energy decomposition at the representative FastVID Keep_1 function.** Fractions use the indicated denominator.

Component	Squared energy
A	5.529×10^{-9}
B	1.907×10^{-5}
C	3.432×10^{-5}
$C - B$	1.525×10^{-5}
$E_{\mathrm{W}} = A + C$	3.433×10^{-5}
$E_{\mathrm{Direct}} = A + B$	1.908×10^{-5}
$\|G\|_F^2$	1.933×10^{-4}

Fraction	Percent
A/E_{W}	0.02
C/E_{W}	99.98
B/C	55.57
$(C-B)/C$	44.43
$(C-B)/E_{\mathrm{W}}$	44.42
$E_{\mathrm{W}}/\|G\|_F^2$	17.76
$E_{\mathrm{Direct}}/\|G\|_F^2$	9.87

At this endpoint, the additional component $C - B$ accounts for 44.42% of total wide-accessible gradient energy and 44.43% of wide-accessible transverse gradient energy.

F.2 Current and optimal coefficient patterns

Define $M = P_\perp GP_H$. For a unit coefficient pattern p in the feature row space, the captured transverse-gradient fraction is

$$c(p) = \frac{\|Mpp^\top\|_F^2}{\|M\|_F^2} = \frac{\|Mp\|_2^2}{\|M\|_F^2}. \tag{F.4}$$

The current one-axis function uses $p = a/\|a\|_2$. The optimal capture using one or k orthonormal coefficient patterns is

$$c_1^\star = \frac{\sigma_1(M)^2}{\|M\|_F^2}, \qquad c_k^\star = \frac{\sum_{j=1}^k \sigma_j(M)^2}{\|M\|_F^2}. \tag{F.5}$$

The oracle patterns are selected after observing the gradient and therefore measure the best local capture available within the fixed feature row space. For an isotropically sampled unit pattern in this q-dimensional space, where $q = \mathrm{rank}(\widetilde{H})$, the expected capture is $1/q$. The optimal-pattern result is derived in Appendix I.5.

F.3 Replication across endpoints

The endpoint analysis covers seeds 43, 45, and 46 under both FastVID and UniComp. Each Wide checkpoint supplies its own label-free correction basis, and all six projected one-axis functions are evaluated on the same 400 held-out videos.

Table F.2. **Transverse gradient capture across six projected one-axis functions.** The upper block reports current and optimal-pattern capture. The lower block reports best-single-pattern remainder under video resampling. Values are percentages of wide-accessible transverse gradient energy.

Condition	Seed	Current	Best 1	Best 2	Best 4
FastVID	43	55.57	72.32	81.91	89.23
FastVID	45	57.20	75.46	82.73	90.28
FastVID	46	54.45	72.92	84.74	91.47
UniComp	43	59.53	73.79	84.02	89.58
UniComp	45	67.46	75.47	83.82	89.91
UniComp	46	39.99	67.86	80.68	88.26

Condition	Seed	Original remainder	Resampled mean	Percentile range
FastVID	43	27.68	29.81	[28.55, 31.24]
FastVID	45	24.54	26.71	[25.52, 28.13]
FastVID	46	27.08	28.94	[27.39, 30.86]
UniComp	43	26.21	28.55	[27.00, 30.95]
UniComp	45	24.53	26.44	[25.13, 27.75]
UniComp	46	32.14	35.33	[32.47, 38.15]

Across these six functions, the best-single-pattern remainder $1 - c_1^\star$ ranges from 24.53 to 32.14%. Its mean is $26.44 \pm 1.67\%$ under FastVID and $27.62 \pm 4.00\%$ under UniComp, where the $\pm$ values denote sample standard deviations across initialization seeds. The resampled means differ from the original remainders by 1.86 to 3.19 percentage points. These resampling ranges are used as a sensitivity check rather than as population confidence intervals.

F.4 Baseline training checkpoints

We evaluate projected one-axis functions at 500, 1,000, 2,000, and 4,000 updates along the baseline-rate uncompressed and FastVID trajectories. Each checkpoint uses a correction direction fitted on 160 videos and the same 400 held-out videos for gradient evaluation. We measure the fraction of total wide-accessible gradient energy outside the current one-axis learning space,

$$\rho_{\text{extra}} = \frac{\|\Pi_{\text{W}} G - \Pi_1 G\|_F^2}{\|\Pi_{\text{W}} G\|_F^2}. \tag{F.6}$$

Unlike the best-single-pattern capture in Appendix F.2, ρ_{extra} uses the current coefficient pattern and includes both along-axis and transverse energy in the denominator. Raw and globally centered Wide corrections have $d_{99} = 1$ at every measured checkpoint. The extra fraction remains between 37.1 and 39.3% without compression and between 51.3 and 53.5% under FastVID, as shown in Figure F.1.

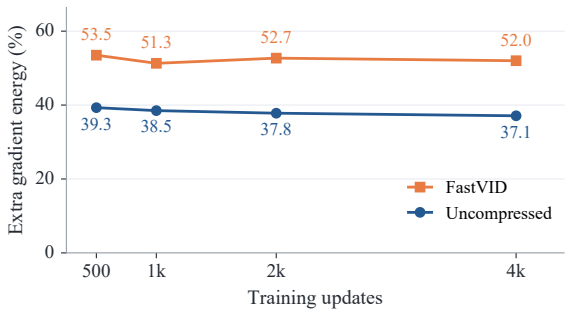

Figure F.1. **Loss-relevant freedom persists during training.** Fraction of wide-accessible gradient energy outside the current one-axis space. Raw and centered corrections have $d_{99} = 1$ at every checkpoint.

Despite this early concentration, the dominant direction continues to evolve. Its absolute cosine similarity to the final direction rises from about 0.76 at 500 updates to about 0.98 by 2,000 updates in both trajectories.

F.5 Continuation from the shared correction

Each shared-rate FastVID Wide checkpoint supplies its own Keep_1 function. From that function, we continue either a Wide lift or Direct while holding the backbone and hidden features fixed. The Wide lift updates its unrestricted affine readout, whereas Direct updates its axis and scalar head. All runs replay the first 500 examples of the original fixed training stream and use fresh AdamW optimizers.

To preserve exact equality at the initial function, we parameterize the trainable correction as

$$\delta_\theta(z) = \delta_{\mathrm{Keep}_1}(z) + [f_\theta(z) - f_{\theta_0}(z)].$$

Here f_{θ_0} is the frozen initial copy of the corresponding readout, initialized to the shared Keep_1 function as in Appendix I.1. The bracketed term is zero at initialization, so both readouts begin from exactly the same correction. Each 500-update continuation run restarts the learning-rate schedule at 1% of the peak rate, warms up linearly for 100 updates, and then follows cosine decay over the remaining 400 updates, ending at 10% of the peak rate. The continuation runs reported here use the shared peak rate 3×10^{-4} for both readouts. Other optimizer settings follow Appendix B.2.

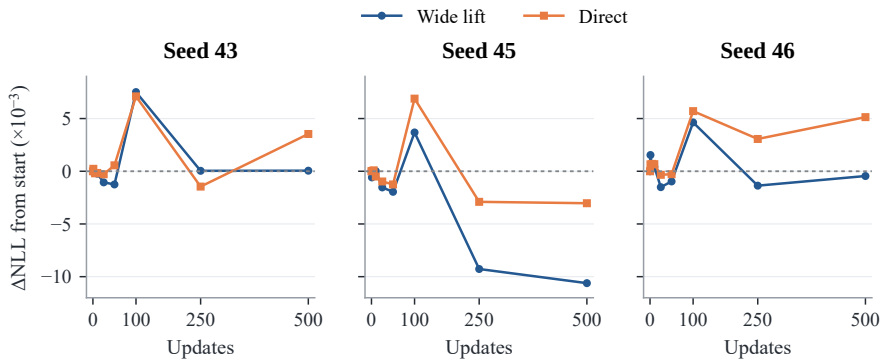

Figure F.2. **Fixed-feature continuation from the shared one-axis correction.** Development NLL relative to the shared Keep_1 start under FastVID at the shared rate. Hidden features remain fixed.

At the shared rate, the early comparisons are mixed. At 10 updates, all paired intervals include zero, and at 100 updates, only seed 45 has an interval for Direct minus Wide lift that is entirely positive. At 500 updates, the Wide lift reaches lower development NLL than Direct in all three runs.

Table F.3. **Continuation NLL contrasts between Direct and the Wide lift.** Values are Direct minus Wide lift at the shared rate, in 10^{-3} NLL units. Early intervals are Bonferroni-adjusted 97.5%. The 500-update intervals are paired 95%.

Comparison	Seed	ΔNLL	Interval
10 updates	43	0.066	$[-0.862, 1.015]$
	45	-0.517	$[-1.398, 0.413]$
	46	0.250	$[-0.655, 1.174]$
100 updates	43	-0.405	$[-2.320, 1.558]$
	45	3.210	$[0.629, 5.925]$
	46	1.073	$[-1.289, 3.463]$
500 updates	43	3.481	$[0.823, 6.197]$
	45	7.582	$[4.734, 10.472]$
	46	5.587	$[2.922, 8.269]$

At 500 updates, Direct has $(5.55 \pm 2.05) \times 10^{-3}$ higher development NLL than the Wide lift on average across the three runs.

F.6 PROJECTING THE FINAL CONTINUATION CORRECTION

The final projection tests whether the small components of the continued correction contribute to likelihood. For each 500-update shared-rate Wide lift, we fit a new dominant correction direction on 160 separate videos and evaluate the resulting projection on the same development set used for continuation.

Table F.4. **Shared start and 500-update continuation endpoints.** Spectra describe the final Wide-lift correction, and NLL is evaluated on the development set. Final Keep_1 projects each final Wide lift onto its newly fitted dominant direction.

Seed	Raw E_1 (%)	Raw d_{99}	Initial Keep_1	Final Wide lift	Final Keep_1	Final Direct
43	99.56	1	1.19629	1.19635	1.20155	1.19983
45	99.77	1	1.19785	1.18724	1.19523	1.19482
46	99.88	1	1.19610	1.19564	1.20182	1.20123
Mean			1.19675	1.19308	1.19953	1.19863
Sample SD			0.00096	0.00507	0.00373	0.00337

All three final Wide lifts have raw $d_{99} = 1$, with more than 99.5% of raw correction energy in the dominant direction. Nevertheless, projecting each final Wide lift onto that direction raises development NLL. The mean projection cost is 0.00646 ± 0.00142 across the three runs.

Table F.5. **Final-projection NLL contrasts.** Values are in 10^{-3} NLL units. Paired 97.5% intervals adjust for the two comparisons within each seed.

Comparison	Seed	ΔNLL	Adjusted 97.5% interval
Final Keep_1 minus final Wide lift	43	5.205	$[2.802, 7.622]$
	45	7.993	$[5.008, 10.960]$
	46	6.173	$[3.952, 8.430]$
Final Keep_1 minus initial Keep_1	43	5.259	$[-0.208, 10.559]$
	45	-2.621	$[-7.151, 1.670]$
	46	5.720	$[0.495, 11.111]$

The adjusted projection-cost intervals remain positive in all three runs. Final projection also removes the point-estimate advantage of the Wide lift over Direct. Final Keep_1 has slightly higher development NLL than final Direct in every run, with paired 95% intervals that include zero.

Table F.6. **Reference NLL contrasts at the 500-update endpoints.** Values are in 10^{-3} NLL units with paired 95% intervals.

Comparison	Seed	ΔNLL	Paired 95% interval
Final Keep_1 minus final Direct	43	1.724	$[-0.344, 3.699]$
	45	0.411	$[-1.574, 2.369]$
	46	0.586	$[-1.882, 3.053]$
Final Wide lift minus initial Keep_1	43	0.054	$[-5.130, 5.020]$
	45	-10.614	$[-15.624, -5.749]$
	46	-0.453	$[-5.232, 4.275]$
Final Direct minus initial Keep_1	43	3.535	$[-0.516, 7.514]$
	45	-3.032	$[-6.738, 0.598]$
	46	5.134	$[0.942, 9.283]$

The retained axis is fitted to minimize correction reconstruction error rather than task NLL. This experiment therefore evaluates whether the dominant correction direction preserves the fitted Wide-lift behavior, not whether it is the best possible one-axis task solution.

F.7 DIRECTLY TRAINED ONE-AXIS FUNCTIONS

The directly trained analysis uses the three FastVID Direct endpoints trained for 4,000 updates at the shared peak rate 3×10^{-4}. Each correction is one-axis by construction. An unrestricted readout with output weight matrix $u\,v^\top$ and bias $u\beta$ implements exactly the same function with the same hidden features. The shared function is the trained Direct correction, not a projection of a Wide endpoint.

We evaluate the completion gradient on the same 400 held-out videos used for the projected endpoints. These examples contain 1,200 supervised completion tokens and 16,808 visual-token observations. The objective averages supervised-token losses within each example and then across examples. With the constant bias row included, the hidden-feature matrix has numerical rank $q = 65$ in all three runs.

Table F.7. **Transverse gradient capture at directly trained one-axis functions.** Shared-rate FastVID endpoints. Values are percentages of wide-accessible transverse gradient energy. Best k uses k optimal coefficient patterns. Remainder is 100 minus Best 1.

Seed	Current	Best 1	Best 2	Best 4	Best 8	Remainder
43	71.90	82.72	88.69	93.72	96.41	17.28
45	72.69	81.08	87.29	93.07	96.22	18.92
46	71.39	82.35	87.46	93.51	96.54	17.65
Mean	71.99	82.05	87.82	93.43	96.39	17.95
Sample SD	0.66	0.86	0.76	0.33	0.16	0.86

The current coefficient patterns capture 71.39 to 72.69% of wide-accessible transverse gradient energy. The optimal single patterns raise capture to 81.08 to 82.72%, yet still leave 17.28 to 18.92% uncaptured. Allowing two, four, and eight patterns progressively closes the gap. The learned pattern is already close to the optimal single pattern, with absolute cosine similarities between 0.93 and 0.95.

Table F.8. **Normalized gradient-energy decomposition at directly trained one-axis functions.** Shared-rate FastVID endpoints. Values are percentages, with mean $\pm$ sample SD in the final column.

Fraction	Seed 43	Seed 45	Seed 46	Mean $\pm$ SD
$E_{\mathrm{W}}/\|G\|_F^2$	19.366	19.237	17.123	18.575 ± 1.259
$E_{\mathrm{Direct}}/\|G\|_F^2$	13.927	13.984	12.226	13.379 ± 0.999
$E_{\mathrm{extra}}/\|G\|_F^2$	5.439	5.253	4.897	5.196 ± 0.276
$E_{\mathrm{extra}}/E_{\mathrm{W}}$	28.087	27.306	28.597	27.997 ± 0.650
$(C-B)/C$	28.100	27.309	28.609	28.006 ± 0.655

The component outside the current one-axis learning space accounts for $28.00 \pm 0.65\%$ of total wide-accessible gradient energy across the three Direct endpoints. This fraction uses the current coefficient pattern, whereas the best-single-pattern remainder allows that pattern to be replaced. These quantities describe correction-space gradients, not hidden-parameter gradients. Across the six projected and three directly trained one-axis functions analyzed in Section 4, even the best single coefficient pattern leaves 17.3 to 32.1% of wide-accessible transverse gradient energy uncaptured.

G DIRECTION, COEFFICIENTS, AND DECODER RESPONSE

G.1 DECODER-ANALYSIS SETUP

The decoder analysis separates the output direction from the token coefficient pattern. We use a Qwen2.5-VL-7B FastVID checkpoint trained for 4,000 updates at a peak rate of 3×10^{-5}. Its Keep_1 direction and three orthogonal random directions are fixed before evaluation. Random-direction interventions retain the learned token coefficients and correction magnitude while changing only the output direction.

Evaluation uses 512 held-out confirmation videos that are disjoint from the correction-basis and decoder-discovery sets. The fixed margin compares the gold option with the strongest wrong option

selected under Base, with that wrong option held fixed across interventions. Visual-key attention sums attention on visual keys at the final prompt query and averages over layers, heads, and videos.

G.2 Direction controls across checkpoints

We evaluate both the decoder-analysis checkpoint and a shared-rate checkpoint on the same confirmation videos, as shown in Figure G.1. The shared-rate checkpoint uses a peak learning rate of 3×10^{-4}.

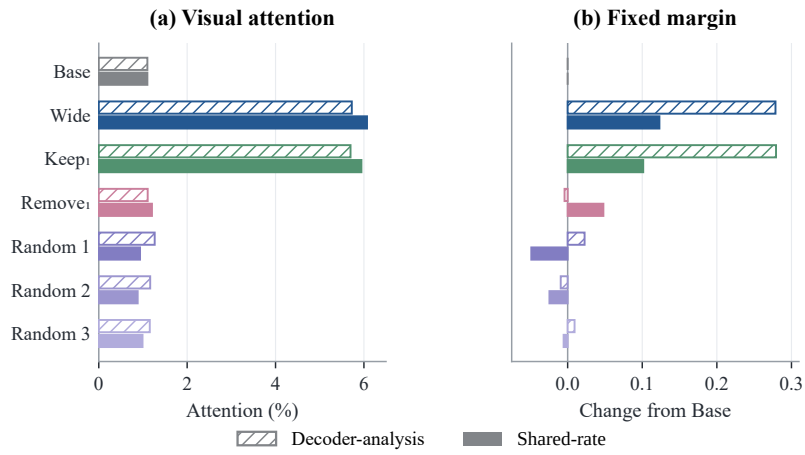

Figure G.1. **Direction controls across checkpoints.** (a) Visual-key attention and (b) Base-relative fixed margin at the decoder-analysis and shared-rate checkpoints. Both use the same 512 confirmation videos.

Table G.1. **Fixed-margin contrasts against random directions.** Keep_1 minus each matched random direction at the decoder-analysis and shared-rate checkpoints. Intervals are Bonferroni-adjusted 98.33%.

Comparator	Decoder-analysis	Shared-rate
Random 1	0.257 [0.144, 0.368]	0.150 [0.020, 0.278]
Random 2	0.289 [0.168, 0.406]	0.126 [−0.005, 0.256]
Random 3	0.270 [0.152, 0.385]	0.107 [−0.024, 0.236]

At the decoder-analysis checkpoint, the fixed-margin contrasts against all three random directions remain positive after adjustment. At the shared-rate checkpoint, only one of the three adjusted intervals is positive. The dominant direction therefore produces a clearer margin response at the decoder-analysis checkpoint, while the visual-key attention response remains similar across the two checkpoints.

On the confirmation set, Keep_1 visual-key attention is 5.70% at the decoder-analysis checkpoint and 5.94% at the shared-rate checkpoint. The corresponding Base-relative margin changes are 0.279 and 0.101. Remove_1 returns visual-key attention close to Base.

G.3 Coefficient-pattern interventions

Within video v, write the one-axis correction as

$$a_j u = \bar{a}_v u + (a_j - \bar{a}_v)u,$$

where $\bar{a}_v$ is the mean token coefficient. Video-mean retains the shared component, Video-centered retains the deviations, and Uniform-RMS replaces all token coefficients in a video by a constant with the same RMS magnitude.

Table G.2. **Coefficient-pattern interventions on MVBench.** Option accuracy and option agreement with Wide at the decoder-analysis checkpoint (%).

Intervention	Opt. acc.	Opt. agree.
Keep_1	52.17	97.77
Uniform-RMS	52.49	85.23
Video-mean	52.22	82.13
Video-centered	50.49	79.35

The shared component carries 72.9% of pooled coefficient energy. Uniform-RMS reaches option accuracy of 52.49%, close to 52.17% for Keep_1, while option agreement with Wide falls from 97.77% to 85.23%. Preserving the output direction and per-video RMS correction magnitude therefore does not preserve the fitted predictions. The token coefficient pattern is itself behaviorally important.

H Additional Breadth and Control Experiments

H.1 InternVL2.5-2B correction structure and retained directions

Compact corrections also appear in InternVL2.5-2B. Under baseline FastVID, the leading three raw directions carry 99.51%, 0.34%, and 0.09% of correction energy. Globally centered and within-video-centered corrections have $d_{99} = 3$, with dominant-direction shares of 96.98% and 97.00%. Centered hidden-feature variation has $d_{99} = 43$.

Table H.1. **InternVL2.5-2B retained-direction accuracy contrasts.** Generated-answer accuracy differences on MVBench under FastVID, in percentage points, with paired 95% intervals.

Comparison	Difference	95% interval
Keep_3 minus Base	+2.98	[1.93, 4.04]
Keep_3 minus Keep_1	+0.63	[0.05, 1.18]
Keep_3 minus Wide	+0.18	[−0.28, 0.63]
Remove_3 minus Base	−0.25	[−0.73, 0.23]

Despite the strong raw concentration, retaining three directions improves generated-answer accuracy over Keep_1 by 0.63 percentage points [0.05, 1.18] and remains close to Wide. This provides a model-family example in which correction energy alone does not determine the retained dimension needed to preserve behavior.

H.2 Architecture and initialization controls

The Qwen2.5-VL-7B FastVID controls test whether concentrated corrections persist under changes to initialization and adapter structure. Reverse-zero moves zero initialization to the other weight matrix, tiny nonzero output uses scale 10^{-6}, and the linear control removes the nonlinear activation. All three use zero down-bias, so the linear control changes both the activation and bias configuration.

Table H.2. **Architecture and initialization controls under FastVID.** MVBench generated-answer accuracy (%), paired 95% gain intervals in percentage points, and correction dimension.

Control	Accuracy	Gain interval	Raw / Centered d_{99}
Reverse-zero	52.63	[0.10, 1.98]	3 / 6
Tiny nonzero output	53.20	[0.45, 2.80]	1 / 1
Linear adapter	53.13	[0.35, 2.77]	1 / 1

All three controls improve generated-answer accuracy over Base while retaining compact corrections. Tiny nonzero output and the linear control have raw and centered $d_{99} = 1$, while reverse-zero has raw and centered d_{99} of 3 and 6.

H.3 NATURAL-LANGUAGE RESPONSES AND LIKELIHOOD

The dominant-direction result also extends to natural-language responses. We evaluate 1,000 training-disjoint examples stratified by answer length and reweighted to the response-length population, with a nested 200-example generation subset. Unlike the correction basis used in the main experiments, the retained axis here is the leading left singular vector of the output weights and captures 94.47% of output-weight energy.

Table H.3. **Natural-language likelihood after one-direction projection.** Population-weighted completion NLL on 1,000 examples, with paired 95% intervals.

Base	Wide	Keep_1	Remove_1
1.430	1.212	1.211	1.430

Contrast	Δ (10^{-3})	95% interval
Wide − Base	−218.099	[−231.123, −205.282]
Keep_1 − Wide	−0.645	[−1.367, 0.056]
Remove_1 − Base	0.112	[−0.656, 0.888]

Population-weighted completion NLL is 1.430 for Base, 1.212 for Wide, 1.211 for Keep_1, and 1.430 for Remove_1. The dominant output-weight direction therefore preserves the likelihood improvement, while removing it returns NLL to Base. Wide and Keep_1 agree on 89.0% of generated answers in the 200-example generation subset.

H.4 NONZERO RANDOM OUTPUT INITIALIZATION

Concentration also emerges when training begins from a broad correction. We initialize all output weights and biases from $\mathcal{N}(0, 0.01^2)$ in Qwen2.5-VL-7B with FastVID and hidden width 64, while matching the hidden initialization of the corresponding zero-output runs for seeds 43, 45, and 46. The Wide adapter is then trained for 4,000 updates with the shared-rate configuration.

Before training, all three corrections have raw $d_{99} = 60$ and globally centered $d_{99} = 61$. After training, raw d_{99} falls to 3 or 4 and centered d_{99} to 4 or 5, with the dominant direction carrying 96.31 to 97.68% of final raw correction energy.

Table H.4. **Correction concentration from nonzero random outputs.** Spectra of the final correction and the change from initialization across three FastVID runs. Energy shares are percentages.

Seed	Raw E_1	Raw d_{99}	Centered E_1	Centered d_{99}
Final correction D_{final}				
43	97.14	3	97.69	4
45	96.31	4	97.07	5
46	97.68	3	97.49	4
Change from initialization $D_{\mathrm{final}} - D_{\mathrm{initial}}$				
43	97.13	3	97.68	4
45	96.30	4	97.04	5
46	97.68	3	97.48	4

The change from initialization has nearly the same spectrum as the final correction, showing that concentration also appears in the learned function change.

The initial development NLL is close to Base, while training lowers the mean NLL to 1.179 across the three runs. Projecting each final correction onto its dominant direction raises NLL in every run, with mean cost 0.0168 ± 0.0019.

Generated-answer accuracy also improves over Base. Mean accuracy is 53.27 ± 0.51% for Wide and 53.02 ± 0.32% for Keep_1, compared with 51.51% for Base. Generated-choice agreement between Wide and Keep_1 ranges from 92.94 to 94.32%.

Table H.5. **Likelihood after nonzero random-output initialization.** Development completion NLL and dominant-direction projection cost across three FastVID runs. Cost is Keep_1 minus Wide. Cost and its paired 95% interval are reported in 10^{-3} NLL units.

Seed	Initial NLL	Wide NLL	Keep_1 NLL	Cost	Paired 95% interval
43	1.46072	1.17581	1.19449	18.685	$[14.012, 23.535]$
45	1.46036	1.18052	1.19551	14.986	$[10.606, 19.804]$
46	1.45929	1.18186	1.19849	16.628	$[11.493, 22.000]$

Table H.6. **Accuracy and prediction agreement after nonzero random-output initialization.** MVBench generated-answer accuracy and Wide–Keep_1 agreement are percentages. Paired accuracy contrasts and 95% intervals are in percentage points.

Seed	Wide accuracy	Keep_1 accuracy	Generated-choice agreement
43	53.77	53.39	94.32
45	52.74	52.82	92.94
46	53.29	52.87	93.32

Seed	Wide minus Base	Keep_1 minus Base	Keep_1 minus Wide
43	$+2.25\ [0.93, 3.58]$	$+1.88\ [0.52, 3.21]$	$-0.38\ [-1.00, 0.23]$
45	$+1.23\ [-0.15, 2.58]$	$+1.30\ [-0.08, 2.65]$	$+0.08\ [-0.60, 0.78]$
46	$+1.78\ [0.40, 3.13]$	$+1.35\ [-0.03, 2.71]$	$-0.43\ [-1.11, 0.25]$

A useful, concentrated correction therefore emerges from broad random outputs, while small additional components still affect likelihood and individual predictions.

H.5 Hidden–readout alignment across compression conditions

The alignment analysis uses shared-rate Qwen2.5-VL-7B checkpoints and the same 400-video gradient-evaluation set with 16,808 visual-token observations. Hidden features are the post-SiLU activations and are globally centered before analysis. These measurements are separate from the correction-basis spectra on 160 videos reported in Table C.4.

Let $H_c \in \mathbb{R}^{r\times n}$ collect the globally centered hidden features. Write their covariance as

$$\Sigma_h = \frac{1}{n} H_c H_c^\top = \sum_j \lambda_j e_j e_j^\top, \qquad \lambda_1 \geq \lambda_2 \geq \cdots, \qquad g_j = \|W_u e_j\|_2^2. \tag{H.1}$$

The unit vectors e_j are hidden principal directions, and g_j is the readout energy gain along mode j. We define the dominant-mode variance share and relative energy amplification by

$$p = \frac{\lambda_1}{\sum_j \lambda_j}, \qquad \kappa = \frac{g_1}{\dfrac{\sum_{j>1} \lambda_j g_j}{\sum_{j>1} \lambda_j}}. \tag{H.2}$$

Thus κ compares the dominant-mode gain with the hidden-variance-weighted mean gain of the remaining modes. The dominant hidden mode contributes a fraction $\lambda_1 g_1 / \sum_j \lambda_j g_j$ of centered correction energy. Substituting p and κ gives ρ in Equation 4. This quantity attributes energy to one hidden principal mode. Centered E_1 instead measures the leading eigenvalue of $W_u \Sigma_h W_u^\top$ divided by its trace. Each mode contributes a positive-semidefinite term, so $E_1 \geq \rho$. Close agreement indicates that the dominant hidden mode accounts for most of the concentration in output space.

Across FastVID and UniComp, centered hidden-feature d_{99} ranges from 13 to 30 and readout d_{99} from 33 to 47. The decomposition in Table H.7 shows that the dominant hidden mode carries 68.74 to 92.81% of centered hidden variance and has relative amplification κ between 9.16 and 20.08. The resulting ρ ranges from 97.08 to 99.62%, while measured centered E_1 ranges from 97.44 to 99.65%. The realized correction is therefore substantially more concentrated than either factor alone.

For each endpoint, we hold H_c fixed and replace W_u by $W_u Q$ for 20 random orthogonal matrices $Q \in \mathbb{R}^{r\times r}$. This preserves both factor spectra while changing their relative alignment. The measured

Table H.7. **Hidden–readout alignment across compression conditions.** p, ρ, and centered E_1 are percentages, while κ is a ratio. Randomized maximum is the largest observed centered E_1 among 20 orthogonal readout rotations.

Condition	Seed	p	κ	ρ	Centered E_1	Randomized max.
FastVID	43	68.74	19.96	97.77	97.91	90.07
	45	80.40	15.77	98.48	98.73	93.73
	46	88.22	18.85	99.30	99.45	97.26
UniComp	43	92.81	20.08	99.62	99.65	97.54
	45	76.42	10.27	97.08	97.44	88.54
	46	87.39	9.16	98.45	99.08	98.08

centered E_1 exceeds the maximum across all 20 randomized alignments at every endpoint. The additional concentration therefore depends on how the hidden representation and readout are aligned rather than on their spectra alone.

The scalar along the dominant hidden mode closely tracks the coefficient pattern of Keep_1 across both compression methods, with Pearson correlations between 0.99675 and 0.99981 across the six runs. Correlations pool token observations after resolving the arbitrary sign of the hidden mode. The concentration also develops rapidly during training. Raw correction d_{99} ranges from 30 to 41 after the first update, falls to 1 or 2 by 500 updates, and ends between 1 and 3.

The representative FastVID trajectory in Figure 3(a) uses seed 43, with the dominant hidden mode evaluated separately at each checkpoint. Relative amplification is $\kappa = 0.592$ and 0.587 after 1 and 20 updates, then reaches 18.47 and 19.96 at 500 and 4,000 updates. Over the first 500 updates, the dominant-mode share of centered hidden variance rises from 11.08% to 89.63%. Run labels 1, 2, and 3 in Figure 3(b) correspond to seeds 43, 45, and 46 under each compression method.

H.6 Affine approximation and one-unit construction

We approximate the scalar along the dominant hidden mode of each shared-rate FastVID Wide adapter by an affine function of RMS-normalized projector tokens. The surrogate is fitted by ordinary least squares to 16,808 visual-token observations from 400 held-out MVBench videos. The fitting set is disjoint from the 512-video development set and is separate from the 160-video correction-basis set.

Video-grouped five-fold cross-validation gives held-out R^2 between 0.967 and 0.978. We refit the coefficients on all 400 videos. For $d = 3584$, the normalized input and affine correction are

$$\bar{z} = \frac{z}{\sqrt{\|z\|_2^2/d + 10^{-6}}}, \qquad \delta_{\mathrm{aff}}(z) = u(w^\top \bar{z} + c). \tag{H.3}$$

Here $h(z)$ denotes the source Wide hidden features, $\bar{h}$ their pooled token mean, and e_1 the leading unit eigenvector of their centered covariance. The vector u is the unit output axis used by Keep_1, fitted on the separate 160-video basis set. We regress $s(z) = e_1^\top (h(z) - \bar{h})$ as $\beta^\top \bar{z} + \beta_0$. With W_u^{W} and b_u^{W} denoting the source Wide readout, the affine coefficients are $w = \gamma\beta$ and $c = \mu_a + \gamma\beta_0$, where $\gamma = u^\top W_u^{\mathrm{W}} e_1$ and $\mu_a = u^\top (W_u^{\mathrm{W}} \bar{h} + b_u^{\mathrm{W}})$.

The final visual token is $z + \delta_{\mathrm{aff}}(z)$. We transfer this approximation into the original Width_1 module by assigning

$$W_d = w^\top, \qquad b_d = c + B, \qquad W_u = u, \qquad b_u = -Bu. \tag{H.4}$$

Its unchanged forward computation then gives

$$\delta(z) = u\ \mathrm{SiLU}(w^\top \bar{z} + c + B) - Bu \approx u(w^\top \bar{z} + c). \tag{H.5}$$

We choose the positive offset by the deterministic rule

$$B = \left\lceil \max\left(0, \sqrt{d}\,\|w\|_2 - c + 20\right) \right\rceil. \tag{H.6}$$

Since $\|\bar{z}\|_2 \le \sqrt{d}$, this rule bounds the preactivation below by 20 in real arithmetic. The offset is determined by the input-norm bound rather than development NLL. The actual offsets and stored hidden biases are given in Table H.8.

The SiLU activation is neither removed nor bypassed. This is an approximate implementation in the original architecture, not an exact finite-precision identity. Constructed and task-trained instances share the architecture but not their initialization or parameter magnitudes. Neither the affine approximation nor constructed Width_1 receives further task-loss optimization.

Table H.8. **One-unit construction offsets and stored FP32 hidden biases.** Biases are shown to four decimals.

Source Wide seed	Offset B	Stored b_d
43	3841	3881.1519
45	5045	5094.5352
46	4872	4924.2637

Table H.9. **Post-training approximation and one-unit construction.** Development NLL under FastVID. Each row uses its corresponding Wide checkpoint.

Seed	Wide	Keep_1	Affine approximation	Constructed Width_1
43	1.17906	1.19629	1.20038	1.20029
45	1.17367	1.19785	1.20269	1.20325
46	1.17911	1.19610	1.20030	1.20028

The constructed adapters remain within 0.001 NLL of their affine approximations in all three runs and retain about 90% of the NLL improvement from Wide over Base. For seed 43, Width_1 trained from initialization with the shared-rate configuration reaches development NLL of 1.40258, whereas the constructed instance reaches 1.20029. The paired difference for constructed minus task-trained Width_1 is -0.20229, with 95% interval $[-0.22080, -0.18362]$.

The construction retains most of the adaptation gain without exactly reproducing Wide. Its advantage over shared-rate task training separates compact representation from the solution reached through learning.

I Proofs

We use real arithmetic and fixed-rank tangent geometry (Vandereycken, 2013) for affine readouts with shared hidden features. Algebraic rank is distinct from numerical rank and from energy-based dimensions such as d_{99}.

I.1 Exact realizability including output bias

Proposition I.1. Exact projected realizability. Every k-direction orthogonal projection of an affine-readout adapter is exactly representable by a k-axis readout with the same hidden computation.

Proof. Let

$$\delta(z) = W_u h_\theta(z) + b_u, \qquad U_k^\top U_k = I_k.$$

Then

$$U_k U_k^\top \delta(z) = U_k\big[(U_k^\top W_u)h_\theta(z) + U_k^\top b_u\big]. \tag{I.1}$$

Thus the projected correction is implemented exactly by retaining h_θ and using directions U_k, coefficient head $U_k^\top W_u$, and bias $U_k^\top b_u$. An unrestricted readout implements the same function with output weight $U_k U_k^\top W_u$ and bias $U_k U_k^\top b_u$. For $k = 1$, Direct uses $v = W_u^\top u$ and $\beta = u^\top b_u$. □

A directly trained k-axis correction $U[V^\top h(z) + \beta]$ is likewise represented exactly by an unrestricted readout with output weight $UV^\top$ and bias $U\beta$. For $k = 1$, this is the Wide lift used for the directly trained Direct endpoints.

I.2 Local learning spaces for rank-k readouts

Let $\widetilde{H} \in \mathbb{R}^{m\times n}$, with $m = r + 1$, include the constant feature row. Write

$$\mathcal{R} = \mathrm{row}(\widetilde{H}), \qquad q = \dim \mathcal{R}.$$

At a shared rank-k function, with the coefficient bias absorbed into V, let

$$D = C\widetilde{H} = UA, \qquad A = V^\top \widetilde{H}, \qquad U^\top U = I_k, \qquad \operatorname{rank}(A) = k.$$

Assume $1 \le k \le \min(d, q)$.

Proposition I.2. Rank-k local learning spaces. With $\widetilde{H}$ fixed, the unrestricted and k-axis readouts have local learning spaces

$$\begin{aligned} \mathcal{T}_{\mathrm{W}} &= \{X \mid \operatorname{row}(X) \subseteq \mathcal{R}\}, \\ \mathcal{T}_k &= \{US + BA \mid \operatorname{row}(S) \subseteq \mathcal{R},\ U^\top B = 0\}. \end{aligned} \tag{I.2}$$

Their dimensions are

$$\begin{aligned} \dim \mathcal{T}_{\mathrm{W}} &= dq, & \dim \mathcal{T}_k &= kq + (d-k)k, \\ \dim(P_\perp \mathcal{T}_{\mathrm{W}}) &= (d-k)q, & \dim(P_\perp \mathcal{T}_k) &= (d-k)k, \end{aligned} \qquad P_\perp = I - UU^\top. \tag{I.3}$$

Hence $\mathcal{T}_k \subseteq \mathcal{T}_{\mathrm{W}}$ with dimension gap

$$(d-k)(q-k),$$

which is positive when $d > k$ and $q > k$.

Proof. For the unrestricted readout,

$$\dot{D} = \dot{C}\widetilde{H}.$$

Varying $\dot{C}$ spans exactly the matrices whose rows lie in $\mathcal{R}$, giving $\dim \mathcal{T}_{\mathrm{W}} = dq$.

For the factorized readout,

$$\dot{D} = \dot{U}A + U\dot{V}^\top \widetilde{H}. \tag{I.4}$$

Write $\dot{U} = U\Omega + B$, where $\Omega^\top = -\Omega$ and $U^\top B = 0$. The term $U\Omega A$ is absorbed into the within-U variation US, while B contributes the transverse term BA. The first component spans kq dimensions. Since A has full row rank, the map $B \mapsto BA$ is injective and contributes $(d-k)k$ dimensions. The two components are orthogonal because $U^\top B = 0$.

Every row of BA lies in $\operatorname{row}(A) \subseteq \mathcal{R}$, which proves $\mathcal{T}_k \subseteq \mathcal{T}_{\mathrm{W}}$. Applying $P_\perp$ removes the within-U component and gives the transverse dimensions in Equation I.3. □

For $k = 1$, this reduces to Proposition 1. Direct transverse changes have the form $ba^\top$. They have rank at most one but span a $(d-1)$-dimensional space. The unrestricted readout can use up to q independent coefficient patterns in a transverse space of dimension $(d-1)q$. Matrix rank and local learning-space dimension are therefore different quantities.

Trainable hidden features. Let A_ϕ denote the coefficient matrix when the coefficient head and hidden computation are allowed to vary. Changes in these quantities contribute $U\dot{A}_\phi$, so

$$P_\perp U \dot{A}_\phi = 0.$$

Hence

$$P_\perp \dot{D} = P_\perp \dot{U} A_\phi. \tag{I.5}$$

Updates to the coefficient head or hidden features therefore introduce no new transverse coefficient pattern at first order for the factorized readout. At the Wide lift $C = UV^\top$, choosing

$$\dot{C} = \dot{U}V^\top + U\dot{V}^\top$$

and using the same hidden-feature differential reproduces every differential of the factorized readout. The unrestricted readout can additionally introduce transverse changes through its readout degrees of freedom.

I.3 Projected gradients and immediate descent

Let P_A project onto $\operatorname{row}(A)$ and P_H onto $\mathcal{R}$. Since $\operatorname{row}(A) \subseteq \mathcal{R}$,

$$P_H P_A = P_A P_H = P_A.$$

For $G = \nabla_D \mathcal{L}$, orthogonal projection onto the two local learning spaces gives

$$\Pi_{\mathrm{W}} G = G P_H, \qquad \Pi_k G = U U^\top G P_H + P_\perp G P_A. \tag{I.6}$$

The within-U and transverse components are orthogonal. The part available to the unrestricted readout but not to the k-axis readout is

$$P_\perp G (P_H - P_A).$$

Therefore

$$\|\Pi_{\mathrm{W}} G\|_F^2 - \|\Pi_k G\|_F^2 = \|P_\perp G (P_H - P_A)\|_F^2. \tag{I.7}$$

For either projected output-space flow $\dot{D} = -\Pi G$, the instantaneous loss derivative is

$$\frac{d\mathcal{L}}{dt} = \langle G, \dot{D} \rangle_F = -\|\Pi G\|_F^2.$$

The difference in Equation I.7 therefore gives the immediate-descent advantage available to the unrestricted readout in this output-space geometry. These output-space flows are not the parameter-space dynamics of AdamW. The empirical analysis uses the projections as a local geometric measure of the loss-relevant changes available to each readout.

I.4 Zero-output initialization

At zero-output initialization, let Wide and Direct share $\widetilde{H}$, let both output heads be zero, and let Direct have unit axis u_0. Write

$$G_0 = \nabla_D \mathcal{L}(0), \qquad K = \widetilde{H}^\top \widetilde{H}.$$

One simultaneous Euclidean gradient step of size η gives

$$D_{\mathrm{W}}^{(1)} = -\eta G_0 K, \qquad D_1^{(1)} = -\eta u_0 u_0^\top G_0 K. \tag{I.8}$$

The output-head gradients are $G_0 \widetilde{H}^\top$ for Wide and $\widetilde{H} G_0^\top u_0$ for Direct, while the axis and hidden-feature gradients vanish at initialization. Multiplying the head updates by the unchanged features yields Equation I.8. This identity describes a matched Euclidean gradient step and need not hold under AdamW preconditioning.

I.5 Optimal-pattern capture and the isotropic reference

Let $M = P_\perp G P_H \neq 0$ and let p be a unit vector in the feature row space. Projection onto transverse changes using coefficient pattern p gives $Mpp^\top$, whose squared norm is $\|Mp\|_2^2$. The Rayleigh quotient therefore gives

$$\max_{\substack{p \in \mathcal{R} \\ \|p\|_2 = 1}} \frac{p^\top M^\top M p}{\|M\|_F^2} = \frac{\sigma_1(M)^2}{\|M\|_F^2}. \tag{I.9}$$

Right singular vectors of M associated with nonzero singular values lie in $\mathcal{R}$ because $M = M P_H$. For $k \leq q$ orthonormal coefficient patterns collected as the columns of S, the captured energy is $\|MS\|_F^2$. Maximizing over such S gives

$$\sum_{j=1}^{k} \sigma_j(M)^2.$$

For an isotropically sampled unit vector p in $\mathcal{R}$, $\mathbb{E}[pp^\top] = P_H/q$. Hence

$$\mathbb{E}[c(p)] = \frac{\mathrm{tr}(M^\top M P_H)}{q \|M\|_F^2} = \frac{1}{q}. \tag{I.10}$$