RESTORING INPUTS IS NOT RESTORING LEARNING: SPARSE RESTORATION IN MULTIMODAL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training on reduced visual views saves computation, yet restoring full images at evaluation need not recover the capability lost during training. Controlled continuations locate a recoverable contribution in the full-minus-reduced gradient at a shared learner state, beyond extra-compute and extra-update controls. Equal-norm remaining errors nevertheless produce opposite short-horizon loss changes. A local finite-horizon expansion explains this distinction through error transport across parameters and optimizer memory, followed by a signed terminal-task projection. GRAFT-V therefore restores the current raw reference-gradient mean with sparse same-state pairs and inverse-probability weighting before one optimizer update, rather than predicting a future task direction. Recovery repeats across six learner configurations; on the primary five-task experiment, fixed sampling closes 92.6% of the 2.44-point gap, while domain-stratified sampling closes 95.9% using 30.8% fewer final-training GPU-hours than reference training. The stronger-view and short-budget comparisons locate the operating boundary: preserving useful evidence directly can cost less than restoring its learning signal.

1 INTRODUCTION

Reducing visual input lowers the cost of multimodal training while allowing the full image to be restored at evaluation (Xing et al., 2025; Singh et al., 2026). This apparent reversibility concerns the input, not what the model learns from it. In Figure 1, reduced-view training remains 2.44 macro points below reference training when both checkpoints receive the same full evaluation input. The omitted evidence is visible again, yet the trained model does not recover the missing capability. What did the reduced view leave unlearned?

Existing approaches improve the retained evidence, the processing path or the gradient estimate. Visual selection preserves useful context under a smaller token budget (Alvar et al., 2025; Yang et al., 2025; Chen et al., 2026c; Song et al., 2026); training-aware methods redistribute computation across layers and stages (Xing et al., 2025; Singh et al., 2026). Multiscale gradient estimation combines inexpensive and expensive views through a paired difference (Ahamed et al., 2026). These mechanisms address efficiency, but choosing an efficient view and explaining what its substitution changes during learning are different questions. Which part of the learned deficit is recoverable through the full view's gradient contribution? Once correction is imperfect, what determines the task effect of the remaining error?

Controlled continuations first locate a recoverable learning contribution. From a common learner state, adding progressively more of the full-minus-reduced gradient closes progressively more of the deficit. Computing and discarding the reference gradient, or inserting dummy work, does not recover the deficit; spending the same time on extra cheap updates yields much less recovery. Thus the benefit is not explained by reference computation or training opportunity alone. At a fixed token count, replacing background tokens with annotated relevant evidence also reduces the deficit, linking this recoverable contribution to visual content rather than merely input size.

Equal-norm remaining errors can produce opposite short-horizon loss changes. Later updates transport the error through parameters and optimizer memory; the terminal loss selects a signed projection onto its parameter-dependent task. A local finite-horizon expansion retains the direction and cancellation lost in norm bounds. State exchanges expose the receiving history, while an exact quadratic example separates the linear response from its finite remainder.

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.

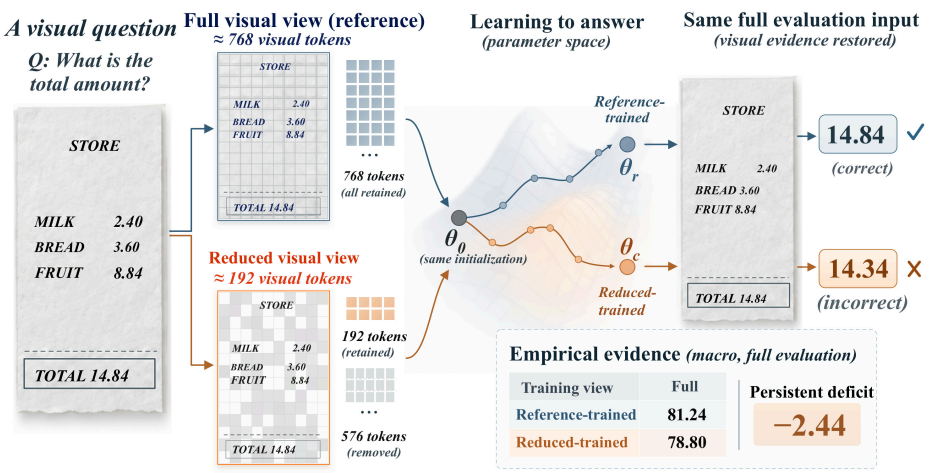


Figure 1: **Restoring the input does not restore acquired capability.** Both trained checkpoints receive the full reference view, yet reduced-view training retains a 2.44-point five-task macro deficit after 12,000 updates. The inset gives eight-seed measured means; the receipt, answers and parameter paths are illustrative, not model outputs.

GRAFT-V restores the current reference contribution instead of estimating its future task direction. Sparse same-state pairs, established inverse-probability weighting and one assembled update recover the raw reference-gradient mean. Across six learner configurations, both fixed and domain-stratified sampling recover most of the aggressive-view deficit. On the primary five-task experiment, fixed sampling closes 92.6% of the gap; stratification raises recovery to 95.9% at 30.8% lower final-training cost than reference training. The equal-budget comparison and stronger-view interventions then locate the useful regime: efficiency depends on what evidence the view retains, what learning signal it displaces and how much opportunity the budget leaves to learn.

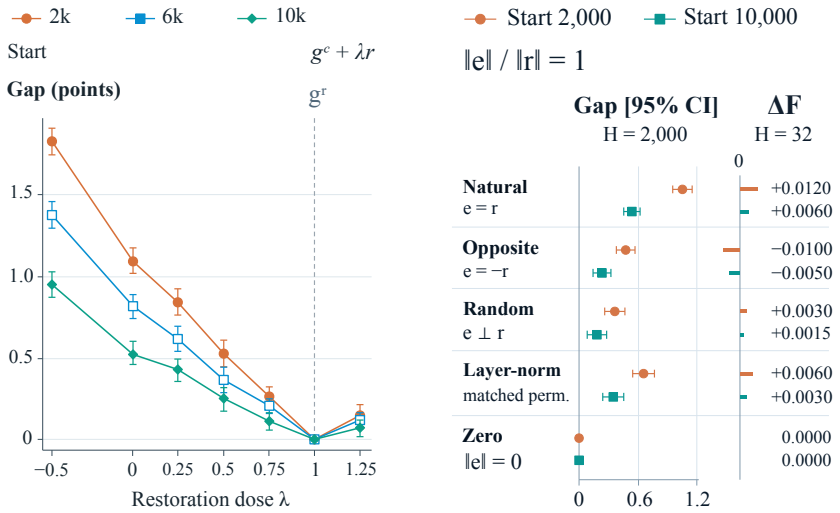
2 What Does a Reduced Training View Leave Behind?

2.1 Prior Work and the Unresolved Questions

Visual reduction ranges from diversity-aware selection and contextual aggregation to prior-corrected attention and structured redundancy control (Alvar et al., 2025; Yang et al., 2025; Chen et al., 2026c; Song et al., 2026). Global–local pruning and text-guided merging preserve complementary token information (Li et al., 2025b; Yu et al., 2026); other methods align distributions, estimate sensitivity or rectify pruning distortions (Chen et al., 2026b; Kim et al., 2026; Cho et al., 2026). Cross-layer methods use spectral evolution or hierarchical token schedules (Chen et al., 2026a; Wu et al., 2026). FastV changes inference processing, whereas PyramidDrop and DUET-VLM also reduce training computation (Chen et al., 2024; Xing et al., 2025; Singh et al., 2026). Fast–slow training explicitly addresses training–inference mismatch (Zhang et al., 2026). Our question is which displaced learning contribution is recoverable, rather than whether such mismatch exists.

Inverse-probability and multiscale estimators provide the gradient construction used here (Horvitz & Thompson, 1952; Ahamed et al., 2026); differentiation through optimization provides the trajectory-sensitivity tool (Franceschi et al., 2017). We use these tools to connect a recoverable visual learning signal to its state-dependent, signed task effect. The question is what the established estimator restores in reduced-view learning, and why the remaining discrepancy can help or harm a task.

Matching the current reference gradient in expectation need not reproduce a reference optimizer trajectory. Conversely, a better view can reduce the task deficit without matching every reference update. We therefore distinguish the update-level target from the finite-training outcome. The following interventions test their relation under shared data and learner states: which contribution recovers the acquired deficit, and why can equal-sized remaining errors have different consequences?



(a) Restoration dose (b) Error direction

Figure 2: **Recovery amount does not determine the remaining error's effect.** (a) Paired doses over 2,000 updates at three starts. (b) Matched entering-error norms produce different 2,000-update gaps and separate 32-update loss changes. Circles/squares identify starts 2,000/10,000; orthogonality is to the residual. Bars show 95% paired seed-mean intervals over eight parents.

2.2 Does the Deficit Survive Restoring the Input?

Five calibrated 12,000-update Qwen2.5-VL-3B endpoints are each evaluated through three visual processors over eight seed blocks (Table B.1). Under full evaluation, reduced training reaches 78.80 against the reference-trained 81.24. The reduced and stronger evaluators retain within-column deficits of 1.74 and 2.34 points. Changing the evaluation interface therefore does not remove the difference acquired by the training paths. The following common-start interventions additionally hold learning rate and clipping fixed, separating gradient restoration from per-path calibration.

Let V_r be the full reference processor and V_c its reduced substitute. On the same logical batch B_t, trainable coordinates and answer-loss reduction, define

$$r_t = g^r(\theta_t; B_t) - g^c(\theta_t; B_t), \qquad g^v(\theta; B) = \nabla_\theta \ell(\theta; B, V_v). \tag{1}$$

Equal predictions at one parameter vector need not imply equal derivatives: at $\theta = 0$, the logistic predictor $\sigma(\theta x)$ equals $1/2$ for every x, whereas its cross-entropy gradient for target one is $-x/2$. The training comparison therefore concerns gradient contributions rather than prediction agreement.

For a continuation of H updates from update t, measure the reference-relative capability gap as

$$R_H = M(\theta^r_{t+H}; V_r) - M(\theta^c_{t+H}; V_r). \tag{2}$$

Here M is the unrounded five-task macro. Superscripts on θ denote reference and substituted trajectories, with the latter possibly corrected; superscripts on g denote the visual processor. The paired reference gradient is evaluated on the corrected path, not at a separate reference checkpoint.

2.3 Which Contribution Recovers the Deficit?

Figure 2 separates how much of the gradient contribution is restored from what the remaining error does to a task.

A deterministic dose supplies $g^c_t + \lambda r_t$, recomputing both gradients at the evolving shared state on each batch. From a common parent at update 2,000, half restoration reduces the 2,000-update gap from 1.07 to 0.52; three-quarter restoration leaves 0.24. The same progression recurs at starts 6,000 and 10,000 (Figure 2(a)). Full restoration uses direct-reference execution and has zero difference by construction; partial doses establish recovery before that endpoint. Reversing the correction increases the gap, isolating the direction of the supplied contribution.

The larger recovery requires using the paired contribution: discarding its gradient, inserting dummy computation or taking extra cheap updates does not reproduce the paired result (Section 5.3). Content also matters. At the same 192-token budget, oracle relevant-region swaps reduce TextVQA and ChartQA continuation gaps from 1.20 and 1.35 to 0.47 and 0.54; matched background swaps leave 1.24 and 1.39 (Figure D.1). These answer-informed interventions isolate evidence content; the correction method itself uses no region annotations.

2.4 Does Error Magnitude Determine Its Consequence?

Figure 2(b) fixes each nonzero remaining-error norm to $\|r_t\|$ before the optimizer acts. Natural and opposite errors produce $+0.012$ and -0.010 loss changes in separate 32-update continuations, yet leave 1.07- and 0.46-point capability gaps after 2,000 updates. A residual-orthogonal rotation and a layerwise norm-matched permutation differ again. The comparison matches the entering error, not the final parameter step. Explaining its sign requires following how the learner transforms its direction; the short-horizon smooth loss and the long-horizon benchmark remain distinct outcomes.

3 How Does Remaining Error Change a Task Loss?

A gradient error enters at one state but is judged through a later task. We separate its construction, transport and terminal projection to explain the equal-norm contrast.

3.1 The Intervention Determines the Entering Error

If an intervention adds d_t to the reduced gradient, then

$$\widehat{g}_t = g_t^{\mathrm{c}} + d_t, \qquad e_t = g_t^{\mathrm{r}} - \widehat{g}_t = r_t - d_t. \tag{3}$$

All gradients share the current corrected state. Substitution leaves $e_t = r_t$ and a paired dose leaves $(1-\lambda)r_t$. The direction controls instead construct $\widehat{g}_t = g_t^{\mathrm{r}} - e_t$ at fixed $\|e_t\| / \|r_t\| = 1$. The controlled quantity is the remaining error, not the added correction.

3.2 The Learner State Transports the Error

Removing fresh error does not erase earlier training. In the state factorial, reduced-path parameters retain a 0.39-point gap with reference moments and future reference inputs. Under reference parameters and inputs, exchanging only the first moment yields a 0.09-point gap (Table B.4). These conditional exchanges show why the same present correction can act on different histories; they do not partition the natural deficit into additive components.

Let s_t contain parameters and optimizer memory, let $\delta s_t = s_t^{\mathrm{c}} - s_t^{\mathrm{r}}$, and let $\mathsf{U}_t(s, g)$ perform one logical update. The closed-loop map $\Psi_t(s, e) = \mathsf{U}_t(s, g^{\mathrm{r}}(s; B_t) - e)$ includes the dependence of the reference gradient on the parameters. Around the reference trajectory,

$$\delta s_{t+1} = J_t \delta s_t + G_t e_t + \xi_t, \qquad J_t = D_s \Psi_t(s_t^{\mathrm{r}}, 0), \quad G_t = D_e \Psi_t(s_t^{\mathrm{r}}, 0). \tag{4}$$

G_t injects the error and J_t transports its parameter and memory effects. In particular, $J_t = D_s \mathsf{U}_t + D_g \mathsf{U}_t D_s g^{\mathrm{r}}$ and $G_t = -D_g \mathsf{U}_t$ on the reference trajectory. The gradient-feedback term in J_t retains loss curvature. Locally Lipschitz derivatives make ξ_t quadratic in the state and error increments.

3.3 The Terminal Task Selects a Signed Response

Reindex a common start as 0 and set $T = H$. The smooth reference-view loss $F_{\mathrm{r}}(s)$ depends only on the parameter component; optimizer memory affects it through intervening updates. For $\Delta F_T = F_{\mathrm{r}}(s_T^{\mathrm{c}}) - F_{\mathrm{r}}(s_T^{\mathrm{r}})$, iterating Equation 4 gives

$$\Delta F_T = \sum_{t<T} a_t^\top e_t + \mathcal{R}_T, \qquad a_t = G_t^\top (J_{T-1} \cdots J_{t+1})^\top \nabla_s F_{\mathrm{r}}(s_T^{\mathrm{r}}). \tag{5}$$

Proposition A.1 proves $\mathcal{R}_T = O(\epsilon^2)$ for uniformly $O(\epsilon)$ errors at fixed T, with locally Lipschitz update derivatives and terminal-loss gradient within one smooth selection and clipping regime.

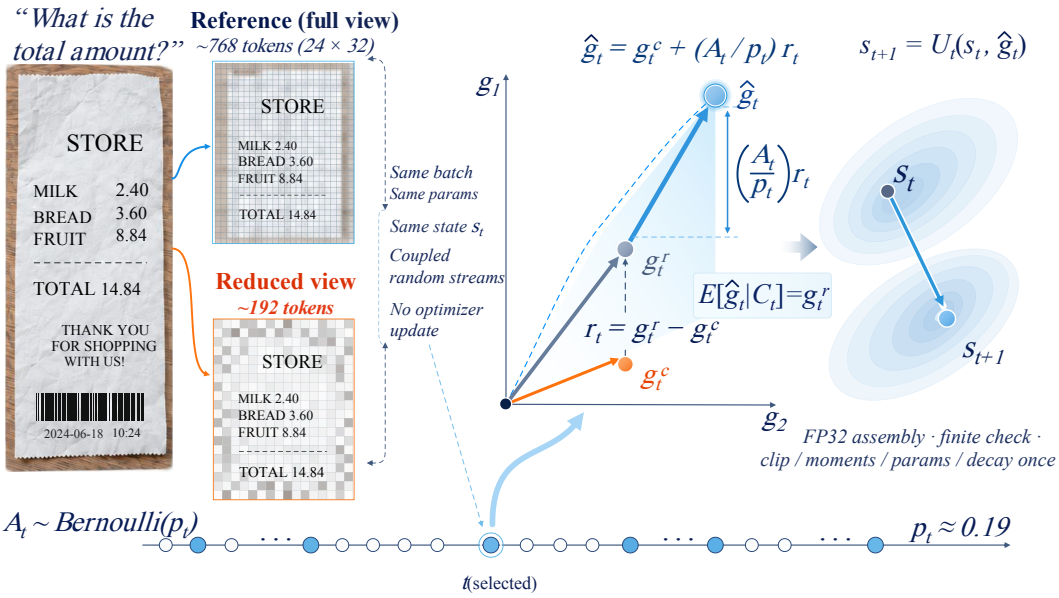


Figure 3: **Sparse restoration at one shared learner state.** The same batch, parameters and coupled random streams yield the paired residual. A Bernoulli draw gates inverse-probability assembly in FP32, followed by a finite check and exactly one clip, moment, parameter and decay update. A selected pulse can overshoot the reference gradient; only its raw conditional mean is restored. The receipt and sampling sequence are schematic.

Corollary A.1 includes state-dependent paired residuals. Near full restoration, the first-order dose response depends on $\sum_t a_t^\top r_t$, rather than the residual norms alone.

Equation 5 concerns a common start. With different initial states, the recursion also transports their inherited displacement, as the state exchanges illustrate. Current-gradient restoration therefore targets new error rather than reversing earlier learning history.

For one small pulse with $a_t \neq 0$, errors $e_t = \pm\epsilon a_t / \|a_t\|$ have equal norm but opposite first-order effects, $\pm\epsilon \|a_t\|$. An equally large adjoint-orthogonal pulse has only a second-order response; across updates, signed projections can cancel. Figure B.1 gives an exact quadratic example. This is why magnitude is insufficient for judging a training-view substitution: the learner and terminal task determine which direction matters. The expansion explains direction and state dependence; the endpoint experiments separately measure the capability attained with finite sparse pulses.

4 Restoring the Current Reference Contribution

The task direction a_t depends on later states and updates. Predicting it would add a separate estimation problem. GRAFT-V instead restores the entire current reference-gradient mean with sparse paired access (Figure 3). This restores every current-information-fixed linear projection in expectation, without selecting one future task direction.

4.1 Sparse Same-State Restoration

Fix an inclusion probability $0 < p_t \leq 1$ before drawing $A_t \sim \mathrm{Bernoulli}(p_t)$. The inclusion draw is conditionally independent of the coupled gradient pair. Assemble

$$\widehat{g}_t = g_t^c + \frac{A_t}{p_t} r_t, \qquad e_t = \left(1 - \frac{A_t}{p_t}\right) r_t. \tag{6}$$

Let $\mathcal{C}_t$ fix state, batch, probability and paired gradients, excluding A_t. Then

$$\mathbb{E}[e_t \mid \mathcal{C}_t] = 0, \qquad \mathrm{Cov}(e_t \mid \mathcal{C}_t) = \frac{1 - p_t}{p_t} r_t r_t^\top. \tag{7}$$

Hence $\mathbb{E}[\widehat{g}_t \mid \mathcal{C}_t] = g_t^{\mathrm{r}}$. Horvitz–Thompson weighting and the paired multiscale decomposition supply this established identity (Horvitz & Thompson, 1952; Ahamed et al., 2026). The matched HT/MGE control shares fixed GRAFT-V's views, draws, arithmetic and optimizer execution.

A mixed-view update selects either g_t^{r} or g_t^{c} and has mean $g_t^{\mathrm{c}} + p_t r_t$, whereas Equation 6 restores the whole difference. Subtraction removes the cheap contribution already counted, and inverse-probability scaling compensates for sparse access. The identity conditions on current information, not on a future adjoint that depends on later draws.

4.2 One Assembled Optimizer Update

Both passes use the same examples, parameters, trainable coordinates, answer-loss reduction and coupled random streams, without an intervening update. FP32 assembly precedes a finite-value check and one global clip, moment update, parameter update and weight-decay operation. Nonfinite assembled gradients skip the optimizer update. At $p_t = 1$, Algorithm 1 runs only the reference pass.

Fixed GRAFT-V uses $p_t = 0.19$: a selected update adds approximately $5.263 r_t$. These finite pulses are evaluated directly, not extrapolated from the small-error theory. Clipping and AdamW moments act on their distribution, so raw-gradient mean restoration is not nonlinear-trajectory unbiasedness (Kingma & Ba, 2015; Loshchilov & Hutter, 2019). The domain-stratified variant uses probabilities $.28, .25, .25, .15, .0879$ for TextVQA, DocVQA, ChartQA, AI2D and GQA. These are fixed within each domain before inclusion. The given domain mixture preserves approximately the same expected call count as uniform sampling. Appendix C.5 states the schedule and distinguishes it from a known-moment variance optimum; no online adaptation is used.

5 When Does Sparse Restoration Help?

We test recovery, construction, transfer and the effect of the training budget.

5.1 Data and Comparison Protocols

The primary learner is Qwen2.5-VL-3B-Instruct with a frozen vision tower, trainable projector and rank-32 language LoRA (Bai et al., 2025b; Hu et al., 2022). Training uses the official partitions of TextVQA, DocVQA, ChartQA, AI2D and GQA, with an image-disjoint calibration holdout. The five official percentage scores form an unrounded macro (Singh et al., 2019; Mathew et al., 2021; Masry et al., 2022; Kembhavi et al., 2016; Hudson & Manning, 2019). Reference processing has a 1,280-token cap and 768-token training median; aggressive DivPrune retains 192 tokens.

Eight seed blocks share initialization, initial optimizer state and ordered examples. We distinguish 12,000-update endpoints, common-start mechanism forks with fixed learning rate and clipping, and 16.55 summed GPU-hour runs on two RTX 4090s. Calibrated paths receive the same 12-candidate selection opportunity. All baseline scores use same-data implementations. GPU-hours sum time across devices; calibration and diagnostic costs are recorded separately from final training. Budgeted training stops at the last complete update; Appendix F specifies scoring and optimization.

5.2 Does Sparse Access Recover the Diagnosed Deficit?

Table 1 compares recovery against visual alternatives and computation controls. Fixed GRAFT-V recovers 2.26 of the 2.44 lost macro points at 16.55 rather than 23.80 final-training GPU-hours. All eight paired gains over DivPrune are positive (95% CI: $[2.171, 2.349]$). Reading, documents and charts account for the larger task-level changes (Table G.1). Every sparse update still uses the cheap view; reference processing is only intermittent.

Domain stratification reaches 81.14 at 16.48 GPU-hours, closing 95.9% of the gap. Its 0.08-point gain over fixed sampling has a paired interval of $[0.013, 0.147]$ (Table H.1). Most recovery comes from the paired contribution; distributing access across domains provides a smaller refinement.

Table 1: **Restoring the signal, not merely adding work, recovers the deficit.** Full reference evaluation; eight-seed means. The upper block uses 12,000 updates with equal calibration opportunity. The lower controls isolate extra computation and extra cheap updates. Costs cover final training. Bold/underline rank upper-block non-reference macros.

Training path	Venue	Macro ↑	Updates	GPU-h ↓
Reference	–	81.240	12,000	23.80
DivPrune	CVPR'25	78.800	12,000	13.07
VisionZip	CVPR'25	80.524	12,000	14.05
PyramidDrop	CVPR'25	80.720	12,000	14.93
Balanced pruning	NeurIPS'25	80.744	12,000	14.62
CLSE	ECCV'26	81.044	12,000	14.48
ZOO-Prune	CVPR'26	80.956	12,000	15.63
OTPrune	CVPR'26	80.960	12,000	15.11
DUET-VLM	CVPR'26	80.856	12,000	15.43
PriorTR	ECCV'26	81.058	12,000	14.25
SFPruner	ECCV'26	81.048	12,000	13.91
View mixture	–	80.030	12,000	15.02
GRAFT-V, fixed	–	<u>81.060</u>	12,000	16.55
GRAFT-V, stratified	–	**81.140**	12,000	16.48
Computation and update controls				
Reference gradient discarded	–	78.800	12,000	16.55
Masked dummy computation	–	78.790	12,000	16.62
Extra cheap updates	–	79.280	15,195	16.55

5.3 Could Extra Compute or More Updates Explain the Gain?

A discarded-reference control pays for both gradients but applies only the cheap one; it remains at 78.80. Masked dummy computation reaches 78.79. Spending the fixed method's time on 15,195 cheap updates instead of 12,000 reaches 79.28, still below paired restoration at 81.06 (Table 1). These controls separately vary computation, learning opportunity and the gradient used by the optimizer. Recovery requires using the paired signal, not merely paying for its computation.

View mixture reaches 80.03 and unweighted correction 80.31, despite their expected reference access. Their targets retain only part of the missing contribution, as the mean calculation in Section 4 predicts. The matched HT/MGE implementation instead shares fixed GRAFT-V's complete trajectory. It verifies estimator identity rather than providing an independent algorithmic win (Table C.3).

5.4 Does Recovery Transfer across Learners?

Both variants recover most of the aggressive-view deficit across six learner configurations, each with its own reference and eight paired seeds (Table 2). The configurations cover five model families and two Qwen2.5-VL sizes; stronger views need not rank identically across them. Task vectors and within-learner training-cost ratios are in Table D.2.

Recovery also survives changes in trainable coordinates. In 2,000-update forks, fixed restoration reduces a 0.22-point projector-only gap to 0.045. With projector and language LoRA the gap falls from 1.07 to 0.19; when the final four vision blocks also train, it falls from 1.47 to 0.29 (Table D.3). Thus the recoverable contribution extends beyond one adaptation subspace, while its size depends on which parameters can learn from the visual input.

5.5 Why Do Pairing and Assembly Matter?

The largest recovery requires a reference matched to the current batch and state. Different-batch pairing reaches 79.24 and a 16-update delay reaches 80.22, versus 81.06 for current-state pairing. Removing inverse-probability weighting or cheap-gradient subtraction also weakens recovery (Table C.2). Each alteration changes which contribution is supplied or how much of it is restored.

Table 2: **Equal-update recovery across six learner configurations, each with its own reference and eight paired seeds.** Fixed and Stratified restore the aggressive reduced view; S + fixed restores the stronger view S. The stronger view S is PriorTR (Chen et al., 2026c), except VisionZip (Yang et al., 2025) for LLaVA-1.5. Bold/underline rank non-reference macros within each learner; Table D.2 gives task vectors and costs.

		Uncorrected views		GRAFT-V (Ours) on Reduced		**Corrected S**
Learner	**Ref.**	**Reduced**	**S**	**Fixed**	**Stratified**	**S + fixed**
Qwen2.5-VL-3B	81.240	78.800	81.058	<u>81.060</u>	**81.140**	81.032
Qwen2.5-VL-7B	84.180	82.720	84.050	84.000	**84.090**	<u>84.060</u>
Qwen3-VL-4B	83.740	80.860	**83.600**	83.490	<u>83.580</u>	83.540
InternVL3.5-4B	82.460	80.320	**82.340**	82.180	82.280	<u>82.310</u>
LLaVA-OV-7B	75.120	72.240	**74.910**	74.800	<u>74.890</u>	74.840
LLaVA-1.5-7B	70.900	69.080	70.650	<u>70.740</u>	**70.770**	70.730

Numerical order is part of that construction. Clipping components separately reaches 80.20; FP16 assembly reaches 80.58 with 749 nonfinite events over 96,000 attempts, versus none for primary FP32 assembly. Full FP32 forward computation reaches 81.05 but costs 23.10 GPU-hours. The exact clipping example in Figure C.1 and optimizer-specific continuations in Table C.1 distinguish the raw-mean target from what the optimizer does with its sampled inputs.

5.6 When Is Improving the View Cheaper?

At 12,000 updates, correction adds 2.26 points to DivPrune and 0.47 to VisionTrim, but changes PriorTR and SFPruner by -0.026 and -0.028 while adding reference work (Figure 4(a)). The reduced benefit is consistent with stronger views retaining more of the useful learning contribution. Task vectors expose heterogeneous changes rather than uniform gains (Table C.4).

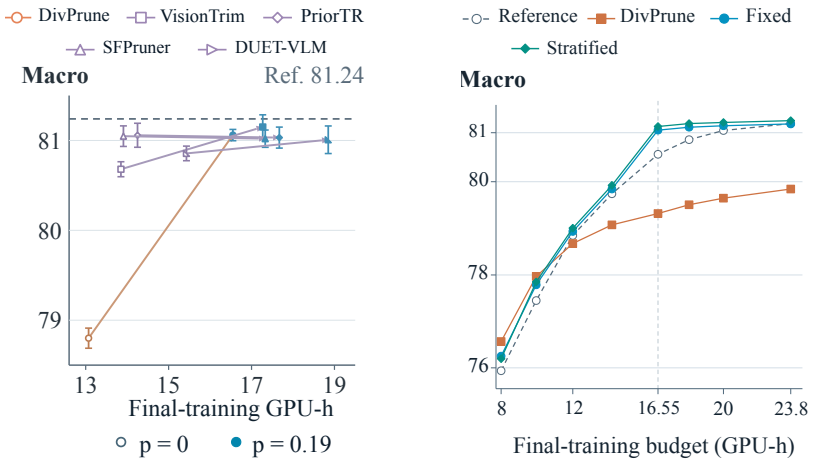
A separate equal-update test asks whether correction can be omitted while reducing cost and preserving capability within margins of 0.20 macro points and 0.50 on every task. SFPruner passes this joint criterion relative to fixed GRAFT-V; PriorTR, VisionTrim and DUET-VLM do not (Table H.4). The test combines all five task requirements with the macro and cost criteria, retaining the original multiplicity adjustment. A nearby mean macro alone cannot certify preservation of every capability. This equal-update decision is distinct from ranking paths at a common time budget.

In the three-task video study, 256 tokens spread across 16 frames reach 66.650 macro at 7.60 training GPU-hours; correcting a four-frame view reaches 66.633 at 9.20. Wider coverage can be cheaper to train. Appendix E separates training costs from caching and reference-frame access.

5.7 Does the Time Budget Change the Decision?

At 16.55 final-training GPU-hours, stratified GRAFT-V reaches 81.146, compared with 81.184 for CLSE, 81.071 for OTPrune and 81.060 for fixed GRAFT-V (Table 3). These nearby endpoints distinguish recovery from dominance over every improved view. The 0.616-point gain over budget-limited reference training combines signal recovery with more completed updates. The reference path completes 8,344 updates, compared with 12,000 for fixed and 12,050 for stratified restoration. Equal-update and equal-time comparisons therefore measure different opportunities to learn.

At 8 and 10 GPU-hours, both corrections trail aggressive reduction. From 12 GPU-hours onward, every sampled corrected endpoint is higher (Figure 4(b)); the reversal shows why budget length matters even with the same sampling construction.



(a) Improve or restore? (b) Available training time

Figure 4: **View quality and training opportunity jointly determine efficiency.** (a) Before/after fixed restoration at 12,000 updates; bars denote 95% seed-mean intervals. Open/filled markers denote zero/.19 inclusion; the dashed line is the reference macro. (b) Valid reference, reduced, fixed and stratified endpoints; lines connect measured budgets, not interpolated crossover estimates.

Table 3: **Capability under a 16.55 GPU-hour training budget.** Full reference evaluation, equal calibration opportunity and eight paired seeds. Bold/underline rank displayed macros; each path stops at its last complete update. PriorTR and SFPruner wrong-setting runs are excluded. Tables G.3–G.4 give task scores and stopping slack.

Training path	Venue	Macro ↑	Seed SD	Updates
Reference	–	80.530	0.130	8,344
DivPrune	CVPR'25	79.280	0.156	15,195
VisionZip	CVPR'25	80.707	0.094	14,135
PyramidDrop	CVPR'25	80.814	0.172	13,302
Balanced pruning	NeurIPS'25	80.866	0.183	13,584
CLSE	ECCV'26	**81.184**	0.146	13,715
ZOO-Prune	CVPR'26	81.045	0.102	12,706
OTPrune	CVPR'26	81.071	0.165	13,143
DUET-VLM	CVPR'26	80.965	0.138	12,871
View mixture	–	80.126	0.120	13,222
GRAFT-V, fixed	–	81.060	0.075	12,000
GRAFT-V, stratified	–	<u>81.146</u>	0.133	12,050

Cheaper updates deliver more progress early, before restoration repays its overhead. With longer budgets, the restored signal changes the capability that can be acquired. These sampled endpoints identify a regime change, not an exact crossover. Together with the equal-update results, they distinguish retaining evidence directly from restoring a displaced learning contribution.

6 Conclusion

Restoring visual input does not automatically restore learning. Controlled interventions locate a recoverable reference contribution, and finite-horizon analysis explains the remaining error through learner-state transport and a signed task projection. GRAFT-V realizes this source-restoration principle with sparse same-state pairs and one assembled update, recovering most of the aggressive-view deficit across learners. Efficiency depends on retained evidence, displaced learning signal and training opportunity; stronger views and short budgets can favor direct evidence preservation.

Reproducibility Statement

Appendix A provides the assumptions and proofs. Appendices C and F describe the paired update, data, training and evaluation protocols. Complete primary scores, paired statistics and training costs are in Appendices G–H. Numerical results accompany the submission; training code and adapted checkpoints are not released. Appendix I states the reproduction scope.

AI Usage Statement

Generative AI assisted with scientific framing, literature discovery and verification, mathematical exposition and proof checking, result interpretation, language revision, illustrative figures, and LaTeX, table and layout preparation. It was not used to generate the experimental observations reported in this paper. Illustrative scenes are distinguished from measured results. The authors take responsibility for the final scientific content and conclusions.

References

Shadab Ahamed, Niloufar Zakariaei, Eldad Haber, and Moshe Eliasof. Multiscale training of convolutional neural networks. *Transactions on Machine Learning Research (TMLR)*, 2026.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

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

Bin Chen, Yuxiang Cai, Yadan Luo, Yi Zhang, Jianwei Yin, and Zhi Chen. Spectral evolution-guided token pruning in multimodal large language models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026a.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 15139 of *Lecture Notes in Computer Science*, pp. 19–35. Springer, 2024. doi: 10.1007/978-3-031-73004-7_2.

Xiwen Chen, Wenhui Zhu, Gen Li, Xuanzhao Dong, Yujian Xiong, Hao Wang, Peijie Qiu, Qingquan Song, Zhipeng Wang, Shao Tang, Yalin Wang, and Abolfazl Razi. OTPrune: Distribution-aligned visual token pruning via optimal transport. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026b.

Zengjie Chen, Yuxiang Cai, Jingcai Guo, Taotao Cai, Jianwei Yin, and Zhi Chen. Accelerating multimodal large language models with prior-corrected token reduction. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2026c.

Hyeonwoo Cho, Donghyeon Baek, Yewon Kim, and Bumsub Ham. Improving visual token reduction via rectifying distortions for efficient multimodal LLM inference. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1165–1173, 2017.

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

D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952. doi: 10.1080/01621459.1952.10483446.

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

Xiaohu Huang, Hao Zhou, and Kai Han. PruneVid: Visual token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19959–19973, 2025. doi: 10.18653/v1/2025.findings-acl.1024.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6700–6709, 2019.

Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max W. F. Ku, Qian Liu, and Wenhu Chen. MANTIS: Interleaved multi-image instruction tuning. *Transactions on Machine Learning Research (TMLR)*, 2024.

Shaobo Ju, Baiyang Song, Tao Chen, Jiapeng Zhang, Qiong Wu, Chao Chang, Huaixi Wang, Yiyi Zhou, and Rongrong Ji. ForestPrune: High-ratio visual token compression for video multimodal large language models via spatial-temporal forest modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 8326–8336, 2026.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 235–251, 2016. doi: 10.1007/978-3-319-46493-0_15.

Youngeun Kim, Youjia Zhang, Huiling Liu, Aecheon Jung, Sunwoo Lee, and Sungeun Hong. ZOO-Prune: Training-free token pruning via zeroth-order gradient estimation in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39572–39582, 2026.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *Transactions on Machine Learning Research (TMLR)*, 2025a.

Kaiyuan Li, Xiaoyue Chen, Chen Gao, Yong Li, and Xinlei Chen. Balanced token pruning: Accelerating vision language models beyond local optimization. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025b. doi: 10.52202/085713-2103.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22195–22206, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, 2022. doi: 10.18653/v1/2022.findings-acl.177.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2200–2209, 2021.

Aditya Kumar Singh, Hitesh Kandala, Pratik Prabhanjan Brahma, Zicheng Liu, and Emad Barsoum. DUET-VLM: Dual stage unified efficient token reduction for VLM training and inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8317–8326, 2019.

Juwon Song, Woohyeong Kim, and Kyeongbo Kong. Structured redundancy modeling for efficient visual token pruning in high-resolution MLLMs. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2026.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, et al. InternVL3.5: Advancing Open-Source Multimodal Models in Versatility, Reasoning, and Efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Hao Wu, Yingqi Fan, Jinyang Dai, Junlong Tong, Yunpu Ma, and Xiaoyu Shen. HiDrop: Hierarchical vision token reduction in MLLMs via late injection, concave pyramid pruning, and early exit. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Conical visual concentration for efficient large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14593–14603, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802, 2025.

Hanxun Yu, Wentong Li, Xuan Qu, Song Wang, Junbo Chen, and Jianke Zhu. VisionTrim: Unified vision token compression for training-free MLLM acceleration. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Dingkun Zhang, Shuhan Qi, Yulin Wu, Xinyu Xiao, Xuan Wang, and Long Chen. Fast-slow efficient training for multimodal large language models via visual token pruning. *arXiv preprint arXiv:2602.03815*, 2026.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data. *Transactions on Machine Learning Research (TMLR)*, 2025.

APPENDIX

A Proof of the Signed Error Response 2
Assumptions, transport and the terminal projection . . . 2

B Mechanism and Training-State Evidence 3
Training and evaluation views . . . 4
Dose and equal-norm directions . . . 4
Parameters, moments and starting context . . . 6

C Construction and Ablations 9
One logical paired update . . . 9
Pairing, numerical order and compute controls . . . 10
Domain sampling and reference allocation . . . 12

D Learners, Subspaces and Relevant Evidence 12
Family-local configurations and full task vectors . . . 12
Matched-budget content interventions . . . 14

E Temporal Coverage at a Fixed Token Budget 15

F Protocols and Reproducibility 17
Data, splits and sampling . . . 17
Optimization and scoring . . . 17

G Complete Primary Results 19
Equal-update results . . . 19
Equal-budget results . . . 19

H Statistics and Resource Accounting 21
Paired uncertainty and task sensitivity . . . 21
Equal-update capability–cost test . . . 22
Training, search and budget endpoints . . . 23

I Scope and Limitations 25

A Proof of the Signed Error Response

We prove the finite-horizon response in Section 3. The closed-loop derivative retains both optimizer memory and the curvature introduced by parameter feedback.

Proposition A.1 (Local response to a gradient-error sequence) *Fix a finite reference trajectory, its initial state, batches and stochastic realizations; perturbed trajectories share that initial state. Suppose the gradient/update maps have locally Lipschitz derivatives and the terminal reference loss has a locally Lipschitz gradient in neighborhoods of the trajectory. For a fixed error sequence u and sufficiently small ϵ,*

$$\Delta F_T(\epsilon; u) = \epsilon D_T(u) + O(\epsilon^2), \tag{A.1}$$

where D_T is linear in u and includes the intervening optimizer updates. The neighborhoods remain within one smooth selection, clipping and finite-update regime.

The proof follows the error into the learner state and then to the terminal loss.

A.1 From the applied gradient to the optimizer state

Let θ denote the trainable parameters and s include those parameters and optimizer memory. On a fixed batch B_t, reference and reduced gradients are derivatives of the same loss under different visual processors,

$$g^v(s; B_t) = \nabla_\theta \ell(\theta; B_t, V_v), \qquad v \in \{\mathrm{r}, \mathrm{c}\}. \tag{A.2}$$

Let $\mathsf{U}_t(s, g)$ perform one logical optimizer update with an externally supplied gradient g, such as either gradient in Equation A.2. Define

$$\Psi_t(s, e) = \mathsf{U}_t(s, g^{\mathrm{r}}(s; B_t) - e), \qquad \mathsf{T}_t(s) = \Psi_t(s, 0). \tag{A.3}$$

At the corrected path's state in Equation A.3, $e_t = g^{\mathrm{r}}(s_t^{\mathrm{c}}; B_t) - [g^{\mathrm{c}}(s_t^{\mathrm{c}}; B_t) + d_t]$, so its actual next state is $\Psi_t(s_t^{\mathrm{c}}, e_t)$. The reference is $\mathsf{T}_t(s_t^{\mathrm{r}})$. The fixed batches and random realizations define a deterministic continuation within convex neighborhoods of one smooth update regime.

The state and error derivatives at $(s_t^{\mathrm{r}}, 0)$ are

$$J_t = D_s \mathsf{T}_t = D_s \mathsf{U}_t + D_g \mathsf{U}_t D_s g^{\mathrm{r}}, \qquad G_t = D_e \Psi_t = -D_g \mathsf{U}_t. \tag{A.4}$$

The chain-rule term includes loss curvature; freezing the reference gradient during state differentiation would omit it.

Lemma A.1 (Local error injection) *For $\delta s_t = s_t^{\mathrm{c}} - s_t^{\mathrm{r}}$, a local derivative-Lipschitz constant L_t^Ψ gives*

$$\delta s_{t+1} = J_t \delta s_t + G_t e_t + \xi_t, \qquad \|\xi_t\| \le \tfrac{1}{2} L_t^\Psi (\|\delta s_t\| + \|e_t\|)^2. \tag{A.5}$$

Proof. Taylor-expand Ψ_t at $(s_t^{\mathrm{r}}, 0)$ in direction $(\delta s_t, e_t)$ and subtract $\Psi_t(s_t^{\mathrm{r}}, 0)$. The linear terms are Equation A.4. Integrating the derivative difference along the line segment bounds the remainder by half its Lipschitz constant times the squared increment in the product sum norm. This proves Equation A.5. □

A.2 From optimizer state to a task loss

For the common initial state, repeatedly substitute Lemma A.1 and write $\Phi_{T,t+1} = J_{T-1} \cdots J_{t+1}$, with an empty product equal to the identity. Then

$$\delta s_T = \sum_{t<T} \Phi_{T,t+1} G_t e_t + \sum_{t<T} \Phi_{T,t+1} \xi_t. \tag{A.6}$$

Let $b_T = \nabla_s F_{\mathrm{r}}(s_T^{\mathrm{r}})$, where F_{r} depends on the parameter part of the state. A terminal gradient Lipschitz constant L_F gives

$$F_{\mathrm{r}}(s_T^{\mathrm{c}}) - F_{\mathrm{r}}(s_T^{\mathrm{r}}) = \sum_{t<T} a_t^\top e_t + \mathcal{R}_T, \quad a_t = G_t^\top \Phi_{T,t+1}^\top b_T, \tag{A.7}$$

$$|\mathcal{R}_T| \le \|b_T\| \sum_{t<T} \|\Phi_{T,t+1} \xi_t\| + \tfrac{1}{2} L_F \|\delta s_T\|^2 . \tag{A.8}$$

Taylor-expanding F_{r} and substituting Equation A.6 proves Equations A.7–A.8. This is the forward/reverse trajectory derivative of Franceschi et al. (2017).

For a fixed forcing sequence u, replace e_t by ϵu_t in Equation A.3 and start at the shared reference state. At a fixed finite horizon, sufficiently small ϵ keeps the trajectory within the assumed neighborhoods. Equation A.5 then gives $\delta s_t = O(\epsilon)$ by induction and $\xi_t = O(\epsilon^2)$. Substituting in Equation A.7 yields

$$D_T(u) = \frac{d}{d\epsilon} F_{\mathrm{r}}(s_T^\epsilon)\Big|_{\epsilon=0} = \sum_{t<T} a_t^\top u_t, \qquad \Delta F_T(\epsilon; u) = \epsilon D_T(u) + O(\epsilon^2). \tag{A.9}$$

Equation A.9 proves Proposition A.1. The corollary below allows the paired residual to vary with the receiving parameters.

Corollary A.1 (State-dependent paired doses) *Under the proposition's assumptions, suppose the paired residual map $r_t(s)$ is locally Lipschitz. Applying correction $d_t = (1-\epsilon) r_t(s_t^\epsilon)$ gives*

$$\Delta F_T(\epsilon) = \epsilon \sum_{t<T} a_t^\top r_t(s_t^{\mathrm{r}}) + O(\epsilon^2). \tag{A.10}$$

Proof. The finite-horizon recursion first gives $\delta s_t = O(\epsilon)$. Local Lipschitz continuity then yields $\epsilon r_t(s_t^\epsilon) = \epsilon r_t(s_t^{\mathrm{r}}) + O(\epsilon^2)$. Substitute this into Equations A.5 and A.7. The finite number of transported remainder terms remains $O(\epsilon^2)$. □

Corollary A.1 connects the dose $\lambda = 1 - \epsilon$ to the local response without freezing the residual. The transported reference-path residual determines the first-order effect near full correction; Figure 2 measures capability over the finite dose range.

For comparison, if $\|J_t\| \le L_t$ and $\|G_t\| \le \kappa_t$, taking norms in Equation A.6 yields

$$\|\delta s_T\| \le \sum_{t<T} \left(\prod_{j=t+1}^{T-1} L_j \right) (\kappa_t \|e_t\| + \|\xi_t\|). \tag{A.11}$$

Equation A.11 bounds state displacement but discards cancellation. Equation A.7 retains the direction needed to explain the loss sign.

A.3 Why source correction needs no future-direction oracle

Fix the current information $\mathcal{C}_t$ and the paired gradient realization. If A_t is a Bernoulli draw independent of that pair conditional on $\mathcal{C}_t$, Equation 6 gives

$$\mathbb{E}[e_t \mid \mathcal{C}_t] = 0, \qquad \mathbb{E}[e_t e_t^\top \mid \mathcal{C}_t] = \frac{1-p_t}{p_t} r_t r_t^\top. \tag{A.12}$$

The identities in Equation A.12 follow from $\mathbb{E}[A_t/p_t] = 1$ and $\mathrm{Var}(A_t/p_t) = (1-p_t)/p_t$. For any $\mathcal{C}_t$-measurable vector a, multiplication of Equation A.12 gives

$$\mathbb{E}[a^\top e_t \mid \mathcal{C}_t] = 0. \tag{A.13}$$

Restoring the vector mean restores every current-information-fixed linear projection without selecting a direction. Sampling covariance can still affect later clipping, moments and feedback; a future adjoint depending on later draws need not be current-information measurable. These established inverse-probability and multifidelity identities (Horvitz & Thompson, 1952; Ahamed et al., 2026) specify the raw-gradient target, not an unbiased nonlinear trajectory.

B Mechanism and Training-State Evidence

The experiments distinguish the evaluation interface, the gradient contribution and the state receiving it. All common-start comparisons clone the full learner state within each of eight parent seed blocks; repeated horizons do not add independent training replicates.

B.1 Training and Evaluation Views

The five calibrated 12,000-update endpoints are each evaluated with native reference processing, 192-token DivPrune and 192-token PriorTR (Table B.1). The comparator within an evaluation group is the reference-trained checkpoint evaluated by that same processor. Calibration permits different selected optimization settings across training paths; the evaluator crossing measures acquired capability, whereas the subsequent forks fix learning rate and clipping across intervention arms.

The reduced-trained endpoint remains below reference training with all three evaluators. Fixed and stratified correction narrow the deficit under full evaluation and exceed reference training when the reduced evaluator is used. Thus restored inputs do not undo training history, and the final processor can still change the ordering of nearby endpoints. The five-task vectors retain this distinction without counting the three evaluations as new training runs.

Table B.1: **Task scores for each training path under native reference, 192-token DivPrune and 192-token PriorTR evaluation.** The three evaluators reuse the same trained endpoints and eight seed blocks.

Training path	Text	Doc	Chart	AI2D	GQA	Macro
Evaluation with Reference						
Reference	81.08	94.70	83.58	82.76	64.08	81.240
DivPrune	77.86	92.08	79.66	81.32	63.08	78.800
GRAFT-V, fixed	80.84	94.54	83.28	82.54	64.10	81.060
GRAFT-V, stratified	80.95	94.63	83.42	82.63	64.07	81.140
PriorTR	80.81	94.47	83.14	82.68	64.19	81.058
Evaluation with DivPrune192						
Reference	78.23	92.95	80.03	82.16	63.53	79.380
DivPrune	75.98	90.94	77.50	80.90	62.88	77.640
GRAFT-V, fixed	78.48	93.13	80.38	81.96	63.83	79.556
GRAFT-V, stratified	78.63	93.23	80.57	82.06	63.82	79.662
PriorTR	78.33	92.94	79.98	82.02	63.78	79.410
Evaluation with PriorTR192						
Reference	80.66	94.42	83.20	82.62	63.96	80.972
DivPrune	77.65	91.90	79.35	81.22	63.05	78.634
GRAFT-V, fixed	80.52	94.36	83.02	82.42	64.02	80.868
GRAFT-V, stratified	80.66	94.46	83.17	82.49	63.97	80.950
PriorTR	80.73	94.43	83.08	82.66	64.17	81.014

B.2 Restoration Dose and Error Direction

For a fixed deterministic dose, the paired gradient obeys

$$g_t^{\mathrm{c}} + \lambda r_t = \nabla_\theta[(1-\lambda)\ell_{\mathrm{c}} + \lambda \ell_{\mathrm{r}}]. \tag{B.1}$$

On $[0, 1]$, this is an objective mixture. Negative doses reinforce the residual and doses above one reverse the remaining error. Both gradients are recomputed at the evolving parameters throughout the 2,000-update continuation. Full restoration takes the direct-reference branch, so its zero gap is an execution identity rather than an estimated gain.

At starts 2,000, 6,000 and 10,000, the uncorrected gaps are 1.07, 0.81 and 0.53. Half restoration leaves 0.52, 0.37 and 0.26, and three-quarter restoration leaves 0.24, 0.21 and 0.12 (Table B.2). Reversal increases the gap; over-restoration rebounds from the zero-gap endpoint. The partial doses locate recoverable capability before the exact full-reference control. Sparse updates use a different distribution of larger, less frequent pulses and are assessed by their own training endpoints.

The direction experiment instead supplies $\widehat{g}_t = g_t^{\mathrm{r}} - e_t$ with matched nonzero entering-error norms. The controls use r_t, $-r_t$, a random vector projected orthogonally to r_t and rescaled to its norm, and a layerwise norm-matched signed permutation. The zero-error arm shares reference execution. Each intervention changes the direction before clipping and moments act; it does not match the resulting parameter displacement.

Table B.2: **Capability recovered by partial paired restoration.** The 2,000-update gap and 95% paired interval at each start use eight seed blocks. At $\lambda = 1$, direct-reference execution gives exactly zero difference and uncertainty.

Dose λ	Start 2,000	Start 6,000	Start 10,000
-0.50	1.800 [1.729, 1.871]	1.360 [1.289, 1.431]	0.930 [0.859, 1.001]
0.00	1.070 [1.003, 1.137]	0.810 [0.743, 0.877]	0.530 [0.463, 0.597]
0.25	0.830 [0.751, 0.909]	0.610 [0.531, 0.689]	0.430 [0.351, 0.509]
0.50	0.520 [0.436, 0.604]	0.370 [0.286, 0.454]	0.260 [0.176, 0.344]
0.75	0.240 [0.177, 0.303]	0.210 [0.147, 0.273]	0.120 [0.057, 0.183]
1.00	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]
1.25	0.130 [0.063, 0.197]	0.110 [0.043, 0.177]	0.080 [0.013, 0.147]

Table B.3 separates 32-update smooth-loss changes from 2,000-update benchmark gaps. At start 2,000, reversing the error changes the local loss response from $+0.012$ to -0.010, yet leaves a positive 0.46-point benchmark gap. The later start preserves this distinction at a smaller scale. Orthogonality here is to the current residual, not to the terminal task adjoint appearing in the theorem.

Table B.3: **Task response at matched entering-error norms, using $g^{\mathrm{r}} - e$.** Orthogonality is to the paired residual, while smooth-loss changes and long-horizon benchmark gaps measure distinct outcomes.

	Entering error	Local loss	2,000-update capability gap		
Error direction	**Norm ratio**	ΔF_{32}	**Gap**	**SD**	**95% interval**
Start 2,000; eight paired seed blocks					
Natural e=r	1.00	+0.0120	1.070	0.090	[0.995, 1.145]
Opposite e=-r	1.00	-0.0100	0.460	0.120	[0.360, 0.560]
Random orthogonal-to-r	1.00	+0.0030	0.370	0.150	[0.245, 0.495]
Layer-norm-matched permutation	1.00	+0.0060	0.650	0.140	[0.533, 0.767]
Zero e=0	0.00	+0.0000	0.000	0.000	[0.000, 0.000]
Start 10,000; eight paired seed blocks					
Natural e=r	1.00	+0.0060	0.535	0.090	[0.460, 0.610]
Opposite e=-r	1.00	-0.0050	0.230	0.120	[0.130, 0.330]
Random orthogonal-to-r	1.00	+0.0015	0.185	0.150	[0.060, 0.310]
Layer-norm-matched permutation	1.00	+0.0030	0.325	0.140	[0.208, 0.442]
Zero e=0	0.00	+0.0000	0.000	0.000	[0.000, 0.000]

B.3 An Exact Directional Example

For $F_{\mathrm{r}}(\theta) = \theta^\top H\theta/2 + c^\top\theta$, with symmetric H, gradient descent has sensitivity $Q = I - \eta H$. A pulse at update t yields

$$\delta\theta_T = \eta Q^{T-t-1} e_t. \tag{B.2}$$

Writing $b = \nabla F_{\mathrm{r}}(\theta_T^{\mathrm{r}})$, its exact loss change is

$$\Delta F_T = \eta b^\top Q^{T-t-1} e_t + \tfrac{1}{2}\eta^2 e_t^\top (Q^{T-t-1})^\top H Q^{T-t-1} e_t. \tag{B.3}$$

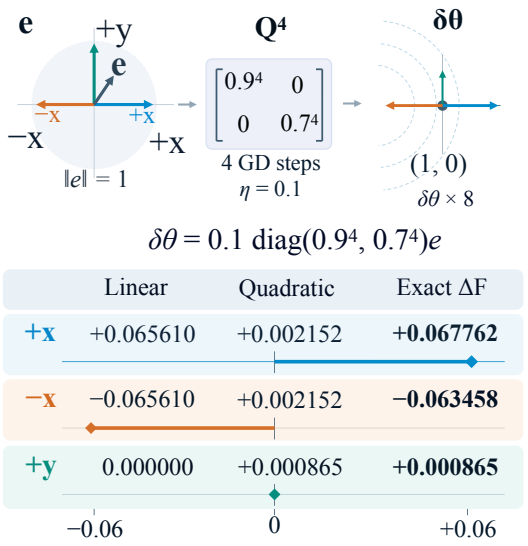


Figure B.1: **Equal-norm pulses can have opposite exact loss effects.** A quadratic objective separates the signed linear projection from the nonnegative second-order term. The horizontal displacement is magnified eightfold in the drawing. These values follow the analytical calculation, not neural measurements.

Figure B.1 uses $H = \mathrm{diag}(1, 3)$, $c = 0$, $\eta = 0.1$, four later gradient steps and reference endpoint $(1, 0)$. Opposite unit pulses on the first coordinate give linear responses ± 0.065610 and the same quadratic contribution, 0.002152. The second-coordinate pulse has no linear contribution but changes finite loss by 0.000865. Equal entering norm therefore fixes neither sign nor magnitude; at a stationary terminal point the linear term vanishes for every direction.

B.4 Parameters and Optimizer Memory

Paired reference and reduced checkpoints at update 6,000 provide the parameter vector, first moment and second moment. Each component is exchanged independently, followed by 2,000 updates with either reference or reduced input. The resulting 16 conditions and two moment-reset controls hold optimizer step age, bias-correction age and the learning-rate schedule fixed. Figure B.2 shows the factorial; Table B.4 gives its intervals.

With reference parameters and inputs, exchanging only the first moment yields a 0.09 gap; exchanging only the second yields -0.015, with an interval spanning zero. Reduced parameters retain a 0.39 gap even with reference moments and inputs. A buffer's effect thus depends on the parameter and input context receiving it. These hybrid-state interventions identify conditional effects rather than additive components of the naturally accumulated deficit.

For momentum, $m_{t+1} = \beta m_t + g_t$ and $\theta_{t+1} = \theta_t - \eta m_{t+1}$ imply

$$\delta m_{t+1} = \beta \delta m_t + [g^{\mathrm{r}}(\theta_t^{\mathrm{c}}; B_t) - g^{\mathrm{r}}(\theta_t^{\mathrm{r}}; B_t)] - e_t. \tag{B.4}$$

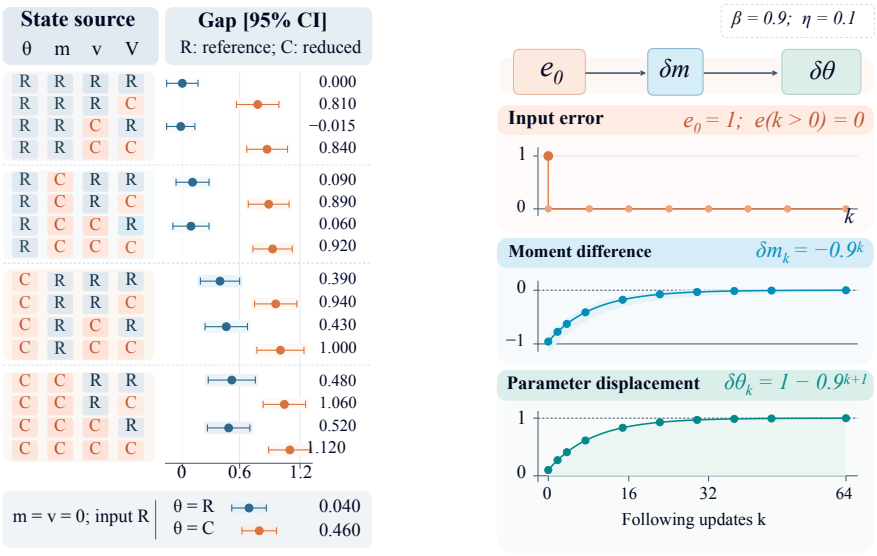
In state order (θ, m), the closed-loop derivatives are

$$J_t = \begin{bmatrix} I - \eta H_t & -\eta\beta I \\ H_t & \beta I \end{bmatrix}, \qquad G_t = \begin{bmatrix} \eta I \\ -I \end{bmatrix}, \tag{B.5}$$

where H_t is the reference-loss Hessian. If later gradient inputs are held fixed, one pulse gives

$$\delta m_{t+1+k} = -\beta^k e_t, \qquad \delta\theta_{t+1+k} = \eta \sum_{j=0}^{k} \beta^j e_t. \tag{B.6}$$

At $\beta = .9$ and $\eta = .1$, Figure B.2(b) shows the decaying moment difference and persistent accumulated parameter displacement. Holding later inputs fixed isolates stored memory; Equation B.5 includes the feedback present when gradients are recomputed.



(a) Receiving-state exchanges (b) Stored momentum

Figure B.2: **Present correction acts on a changed state.** (a) Parameter, moment and input sources crossed at update 6,000 (R/C: reference/reduced), with moment resets below. (b) Exact held-input momentum response: the moment difference decays while parameter displacement persists.

Table B.4: **Parameter, first-moment, second-moment and input sources crossed at update 6,000, followed by 2,000 updates in eight paired blocks.** R/C identify reference/reduced sources; step age, scheduler and bias-correction age are shared, and reset controls zero both moments.

State and continuation sources				2,000-update capability gap		
θ	m	v	**Input**	**Gap**	**SD**	**95% interval**
R	R	R	R	0.000	0.000	[0.000, 0.000]
R	R	R	C	0.810	0.090	[0.735, 0.885]
R	R	C	R	-0.015	0.090	[-0.090, 0.060]
R	R	C	C	0.840	0.110	[0.748, 0.932]
R	C	R	R	0.090	0.090	[0.015, 0.165]
R	C	R	C	0.890	0.110	[0.798, 0.982]
R	C	C	R	0.060	0.110	[-0.032, 0.152]
R	C	C	C	0.920	0.130	[0.811, 1.029]
C	R	R	R	0.390	0.090	[0.315, 0.465]
C	R	R	C	0.940	0.110	[0.848, 1.032]
C	R	C	R	0.430	0.110	[0.338, 0.522]
C	R	C	C	1.000	0.130	[0.891, 1.109]
C	C	R	R	0.480	0.110	[0.388, 0.572]
C	C	R	C	1.060	0.130	[0.951, 1.169]
C	C	C	R	0.520	0.130	[0.411, 0.629]
C	C	C	C	1.120	0.150	[0.995, 1.245]
R (reset)	RESET	RESET	R	0.040	0.100	[-0.044, 0.124]
C (reset)	RESET	RESET	R	0.460	0.100	[0.376, 0.544]

B.5 Starting Context and Training Opportunity

Starts at updates 1,000, 4,000, 8,000 and 10,000 are crossed with horizons 250, 500, 1,000 and 2,000 (Table B.5). Each start changes checkpoint, optimizer history and schedule phase. Later contexts give smaller reduced-view gaps at fixed duration, whereas longer continuations enlarge gaps within a start. Both corrections stay closer to the reference throughout the grid.

The common endpoint at update 12,000 instead changes both context and duration (Table B.6). All non-reference means and 95% intervals are retained, including negative gaps. Shared horizons reuse eight parents; reference-minus-itself gaps are exactly zero.

Table B.5: **All equal-duration continuations.** Each cell gives the reference-relative gap and 95% paired interval over eight parents. The omitted reference-minus-itself condition is identically zero. Horizons within a parent are repeated observations.

Start	H	Reduced	Fixed	Stratified	PriorTR
1,000	250	0.167 [0.118, 0.217]	0.019 [-0.021, 0.059]	0.024 [-0.016, 0.064]	0.015 [-0.025, 0.055]
1,000	500	0.276 [0.224, 0.327]	0.052 [0.012, 0.093]	0.047 [0.006, 0.087]	0.047 [0.006, 0.088]
1,000	1,000	0.540 [0.485, 0.595]	0.113 [0.071, 0.156]	0.049 [0.008, 0.090]	0.105 [0.062, 0.147]
1,000	2,000	0.996 [0.937, 1.056]	0.189 [0.145, 0.233]	0.089 [0.048, 0.131]	0.183 [0.138, 0.228]
4,000	250	0.132 [0.083, 0.180]	0.040 [-0.001, 0.080]	0.006 [-0.034, 0.045]	0.030 [-0.010, 0.071]
4,000	500	0.262 [0.211, 0.313]	0.030 [-0.011, 0.070]	0.027 [-0.014, 0.067]	0.059 [0.017, 0.100]
4,000	1,000	0.454 [0.400, 0.508]	0.085 [0.043, 0.126]	0.065 [0.024, 0.107]	0.079 [0.037, 0.121]
4,000	2,000	0.890 [0.832, 0.949]	0.149 [0.106, 0.193]	0.100 [0.058, 0.142]	0.147 [0.103, 0.191]
8,000	250	0.093 [0.045, 0.141]	0.026 [-0.014, 0.066]	-0.005 [-0.045, 0.034]	0.019 [-0.021, 0.059]
8,000	500	0.194 [0.144, 0.244]	0.051 [0.010, 0.092]	0.012 [-0.027, 0.052]	0.043 [0.003, 0.084]
8,000	1,000	0.376 [0.323, 0.429]	0.057 [0.016, 0.098]	0.046 [0.005, 0.087]	0.056 [0.014, 0.097]
8,000	2,000	0.678 [0.621, 0.734]	0.104 [0.062, 0.147]	0.070 [0.029, 0.112]	0.140 [0.097, 0.184]
10,000	250	0.086 [0.039, 0.134]	-0.002 [-0.041, 0.037]	0.008 [-0.031, 0.048]	0.027 [-0.014, 0.067]
10,000	500	0.167 [0.118, 0.217]	0.019 [-0.021, 0.059]	0.024 [-0.016, 0.064]	0.015 [-0.025, 0.055]
10,000	1,000	0.272 [0.221, 0.323]	0.058 [0.017, 0.099]	0.013 [-0.026, 0.053]	0.051 [0.010, 0.092]
10,000	2,000	0.505 [0.450, 0.559]	0.094 [0.052, 0.136]	0.071 [0.029, 0.112]	0.089 [0.046, 0.131]

Table B.6: **Continuations to update 12,000.** Cells give reference-relative gaps and 95% paired intervals over eight parents; the exact reference-zero row is omitted.

Start	H	Reduced	Fixed	Stratified	PriorTR
1,000	11,000	1.340 [1.278, 1.402]	0.257 [0.212, 0.302]	0.133 [0.090, 0.176]	0.250 [0.204, 0.296]
4,000	8,000	1.168 [1.107, 1.229]	0.193 [0.149, 0.237]	0.122 [0.079, 0.164]	0.224 [0.178, 0.269]
8,000	4,000	0.802 [0.744, 0.860]	0.154 [0.111, 0.197]	0.069 [0.028, 0.110]	0.154 [0.110, 0.198]
10,000	2,000	0.505 [0.450, 0.559]	0.094 [0.052, 0.136]	0.071 [0.029, 0.112]	0.089 [0.046, 0.131]

C Construction and Ablations

C.1 One Logical Paired Update

Algorithm 1 shares one inclusion draw across the logical batch, its microbatches and devices. Reference and reduced buffers use identical answer-loss reductions; the full learner state is unchanged until assembly and the finite-value check finish. The dense branch bypasses the reduced pass. Fixed GRAFT-V, HT fixed and the BF16-forward/FP32-assembly condition refer to this same construction; matched HT/MGE also shares its draws and execution.

Algorithm 1 GRAFT-V for one logical batch

Require: Parameters and optimizer state s_t, logical batch B_t, views $V_{\mathrm{c}}, V_{\mathrm{r}}$, probability $0 < p_t \leq 1$

Freeze s_t during all passes; clear accumulated gradient buffers
if $p_t = 1$ **then**
    Accumulate reference gradient $\widehat{g}_t \leftarrow g^{\mathrm{r}}(s_t; B_t)$
else
    Draw $A_t \sim \mathrm{Bernoulli}(p_t)$ using the inclusion stream
    Accumulate reduced gradient g_t^{c} over the full logical batch
    $\widehat{g}_t \leftarrow g_t^{\mathrm{c}}$
    if $A_t = 1$ **then**
        Reuse the examples, parameters and coupled stochastic realization
        Accumulate reference gradient g_t^{r} with identical loss reduction
        Assemble in FP32: $\widehat{g}_t \leftarrow g_t^{\mathrm{c}} + (g_t^{\mathrm{r}} - g_t^{\mathrm{c}})/p_t$
Unscale once if scaling is enabled; the primary loss scale is 1
if the assembled gradient is nonfinite **then**
    Discard buffers and skip parameter and moment updates
else
    Clip the global assembled gradient once
    Apply one optimizer moment update, parameter update and weight decay

C.2 Raw Means and Nonlinear Updates

Expanding the coordinatewise square of the restored raw gradient yields

$$\mathbb{E}[\widehat{g}_t \odot \widehat{g}_t \mid \mathcal{C}_t] = g_t^{\mathrm{r}} \odot g_t^{\mathrm{r}} + \frac{1 - p_t}{p_t} r_t \odot r_t. \tag{C.1}$$

This pre-clipping identity describes the raw squared input to an Adam second moment. The actual algorithm clips first, so its moments consume $\mathrm{clip}(\widehat{g}_t) \odot \mathrm{clip}(\widehat{g}_t)$. Restoring the raw mean does not commute with this nonlinear operation.

The exact two-outcome example uses $g^{\mathrm{c}} = (.6, .1)$, $g^{\mathrm{r}} = (.6, .3)$ and clip radius .7. At $p = .19$, the selected gradient is $(.6, 1.1526)$, while the unselected gradient is $(.6, .1)$. Their weighted raw mean equals the reference, but the clipped mean is approximately $(.5474, .1990)$ (Figure C.1). The two-outcome calculation isolates the change in mean caused by clipping.

Table C.1 compares 2,000-update capability gaps for momentum SGD and two AdamW second-moment settings, each with its own reference continuation and eight seeds. Increasing inclusion reduces their gaps, with different intermediate responses across optimizers. At $p = 1$, direct-reference execution removes the sampling difference exactly. The reference rate is therefore an operating parameter of the complete learner, not just a multiplier in the raw estimator.

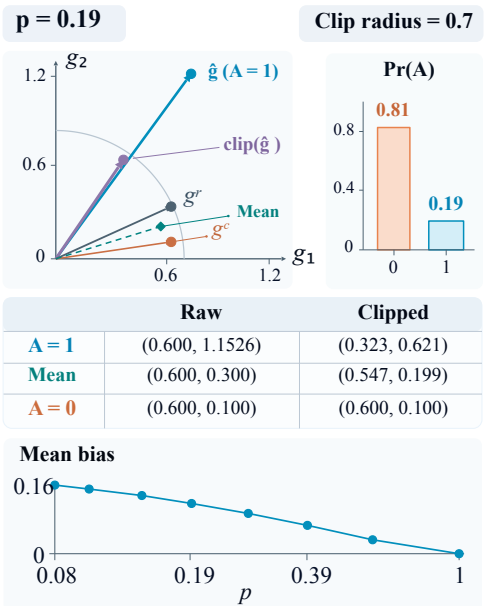


Figure C.1: **Clipping need not preserve a restored mean.** Exact Bernoulli outcomes at $p = .19$ and clip radius .7 give the reference gradient in raw expectation but a different clipped mean. The final trace varies the inclusion probability within the same analytical example.

Table C.1: **Inclusion rate affects different optimizers differently.** Remaining capability gaps after 2,000 updates with each optimizer's own reference and eight seeds. Dense inclusion uses direct-reference execution.

Optimizer	$p = .08$	$p = .19$	$p = .39$	$p = 1$
Momentum SGD, $\beta = .9$	0.330	0.150	0.055	0.000
AdamW, $\beta_2 = .999$	0.440	0.210	0.070	0.000
AdamW, $\beta_2 = .95$	0.480	0.190	0.090	0.000

C.3 Pairing, Arithmetic and Compute Controls

Different-batch pairing, stale residuals and unweighted correction reach 79.24, 80.22 and 80.31, compared with 81.06 for current-state FP32 assembly (Table C.2). Omitting subtraction double-counts the cheap component; clipping each component separately changes the order of nonlinear operations. FP16 assembly produces 749 nonfinite events over 96,000 update attempts, whereas primary FP32 assembly has none. Full FP32 forward computation costs 23.10 GPU-hours while reaching 81.05. Reference-signal restoration and numerical precision therefore have different empirical costs.

Discarded reference gradients, dummy computation and extra cheap updates isolate alternative explanations for recovery (Table C.3). The first two pay extra computational cost without applying the paired signal; the third converts the fixed method's time into more learning opportunities. View mixture and unweighted correction instead alter the mean gradient target. Matched HT/MGE is the exact estimator-identity control, not a separate successful training replicate.

C.4 Improving the View before Correcting It

The equal-update comparison applies fixed restoration to each operator while retaining the reference target (Table C.4). Gains of 2.260, 0.470 and 0.152 for DivPrune, VisionTrim and DUET-VLM contrast with changes of -0.026 and -0.028 for PriorTR and SFPruner. The small negative mean changes have heterogeneous task components; they locate settings where extra reference work brings little marginal benefit. The task-aware noninferiority test in Appendix H.3 separately assesses capability and cost at equal updates.

Table C.2: **Pairing, arithmetic and precision ablations.** Nonfinite counts cover eight 12,000-update runs, including skipped attempts that consume computation. Direct reference uses the dense fast path.

Construction	Macro	Nonfinite / 96k attempts	GPU-h
Paired BF16/FP32 assembly	81.060	0	16.55
No inverse probability	80.310	0	16.55
Different reference batch	79.240	0	16.56
Residual delayed by 16 updates	80.220	0	16.43
No cheap subtraction	79.810	0	16.55
Clip-before-assembly	80.200	0	16.55
FP16 assembly	80.580	749	16.50
All FP32	81.050	0	23.10
Direct reference	81.240	0	23.80

Table C.3: **Computation controls for paired restoration.** Discarded reference gradients, dummy passes and extra cheap updates separate the gradient target from computational work. Reference calls are inclusion expectations.

Control	Macro	GPU-h	Ref. calls	Updates
Cheap	78.800	13.07	0	12,000
Reference backward discarded	78.800	16.55	2280	12,000
Masked dummy tokens	78.790	16.62	0	12,000
Extra cheap updates	79.280	16.55	0	15,195
View mixture	80.030	15.02	2280	12,000
No IP weighting	80.310	16.55	2280	12,000
HT fixed	81.060	16.55	2280	12,000
Matched MGE	81.060	16.55	2280	12,000

Table C.4: **Restoration applied to different visual operators at 12,000 updates, with reference inclusion zero or 0.19.** Task scores distinguish heterogeneous changes behind the macro gain and added training cost.

		Task capability					Aggregate and cost	
Operator	p	**Text**	**Doc**	**Chart**	**AI2D**	**GQA**	**Macro**	**GPU-h**
DivPrune	0.00	77.86	92.08	79.66	81.32	63.08	78.800	13.07
DivPrune	0.19	80.84	94.54	83.28	82.54	64.10	81.060	16.55
VisionTrim	0.00	80.08	94.20	82.62	82.20	64.30	80.680	13.85
VisionTrim	0.19	80.95	94.62	83.47	82.62	64.09	81.150	17.27
PriorTR	0.00	80.81	94.47	83.14	82.68	64.19	81.058	14.25
PriorTR	0.19	80.84	94.53	83.23	82.56	64.00	81.032	17.67
SFPruner	0.00	80.68	94.50	83.22	82.72	64.12	81.048	13.91
SFPruner	0.19	80.78	94.49	83.19	82.58	64.06	81.020	17.33
DUET-VLM	0.00	80.37	94.39	83.04	82.48	64.00	80.856	15.43
DUET-VLM	0.19	80.69	94.53	83.20	82.52	64.10	81.008	18.85

C.5 The Evaluated Domain Schedule

Uniform inclusion uses $p = .19$ on every domain. Stratification uses fixed probabilities $.28, .25, .25, .15, .0879$ for TextVQA, DocVQA, ChartQA, AI2D and GQA, respectively (Table C.5). With domain mixture $(.18, .18, .16, .20, .28)$, its expected 12,000-update call count is 2,280.14, close to the uniform 2,280. The measured macro and training cost are 81.140 and 16.48 GPU-hours, compared with 81.060 and 16.55 for uniform inclusion. These probabilities define the evaluated schedule; the method makes no claim of an adaptive or numerically optimal allocation.

Table C.5: **Fixed domain-inclusion schedules.** Domain frequencies are $(.18, .18, .16, .20, .28)$ in the displayed order. Probabilities are fixed within each domain before inclusion.

Schedule	TextVQA	DocVQA	ChartQA	AI2D	GQA
Uniform	.1900	.1900	.1900	.1900	.1900
Stratified	.2800	.2500	.2500	.1500	.0879

The distinction between call count and weighted reference cost can be expressed for domain weights w_h, residual second moments $\nu_h = \mathbb{E}[\|r_t\|^2 \mid h]$ and costs $c_h > 0$ as

$$C_{\text{call}} = \sum_h w_h p_h, \qquad C_{\text{ref}} = \sum_h w_h c_h p_h. \tag{C.2}$$

For fixed known moments, minimizing $\sum_h w_h \nu_h (1 - p_h)/p_h$ subject to a fixed C_{ref} gives

$$-\frac{w_h \nu_h}{p_h^2} + \mu w_h c_h = 0, \qquad p_h = \sqrt{\frac{\nu_h}{\mu c_h}}. \tag{C.3}$$

With feasible probability bounds, the conditional solution is

$$p_h^* = \min\{p_{\max}, \max\{p_{\min}, \sqrt{\nu_h/(\mu c_h)}\}\}, \tag{C.4}$$

where μ meets the budget. A call-only constraint sets $c_h = 1$. This known-input calculation characterizes raw-gradient variance, not the finite-training capability of the evaluated schedule or an implemented optimality guarantee.

D Learners, Subspaces and Relevant Evidence

D.1 Family-Local Implementations

The six configurations cover Qwen2.5-VL at two sizes, Qwen3-VL, InternVL3.5, LLaVA-OneVision and LLaVA-1.5 (Bai et al., 2025b;a; Wang et al., 2025; Li et al., 2025a; Liu et al., 2024). Each uses projector and rank-32 language LoRA adaptation, 12,000 updates and eight paired seeds. These comparisons use two H200 GPUs; cost ratios are normalized within each learner, not converted into the primary RTX 4090 GPU-hours.

Table D.1 identifies the reference and reduced budgets and the stronger-view alternative. PriorTR is used in the current-family configurations, with its selection interface ported where needed; LLaVA-1.5 uses VisionZip. Each implementation retains its model's processor and position handling. Table D.2 gives all 36 task vectors and within-learner cost ratios behind Table 2, preserving changes that differ between learners and tasks.

Table D.1: Family-local visual configurations using PriorTR (Chen et al., 2026c) or VisionZip (Yang et al., 2025). Budgets count visual tokens entering the language model; every suite uses eight paired seeds on two H200 GPUs.

Learner	**Reference cap**	**Reduced K**	**Stronger view**
Qwen2.5-VL-3B	1280	192	PriorTR
Qwen2.5-VL-7B	1280	256	PriorTR
Qwen3-VL-4B	1280	192	PriorTR
InternVL3.5-4B	1280	192	PriorTR port
LLaVA-OneVision-7B	1280	128	PriorTR port
LLaVA-1.5-7B	576	64	VisionZip

Table D.2: **Task-level recovery with each family's own reference.** Cost ratios are local to the H200 family study, rather than conversions of RTX 4090 hours; Table D.1 gives the operators and token budgets.

	Task capability					Aggregate and cost	
Path	**Text**	**Doc**	**Chart**	**AI2D**	**GQA**	**Macro**	**Cost ratio**
Qwen2.5-VL-3B; eight paired seeds; family-local costs							
Reference	81.08	94.70	83.58	82.76	64.08	81.240	1.000
Reduced	77.86	92.08	79.66	81.32	63.08	78.800	0.560
Stronger	80.81	94.47	83.14	82.68	64.19	81.058	0.680
Fixed	80.84	94.54	83.28	82.54	64.10	81.060	0.700
Stratified	80.95	94.63	83.42	82.63	64.07	81.140	0.700
Strong+fixed	80.84	94.53	83.23	82.56	64.00	81.032	0.840
Qwen2.5-VL-7B; eight paired seeds; family-local costs							
Reference	84.90	95.80	87.50	86.60	66.10	84.180	1.000
Reduced	83.18	94.47	85.22	85.39	65.34	82.720	0.560
Stronger	84.75	95.68	87.30	86.49	66.03	84.050	0.680
Fixed	84.69	95.64	87.22	86.45	66.01	84.000	0.700
Stratified	84.79	95.72	87.36	86.53	66.05	84.090	0.700
Strong+fixed	84.76	95.69	87.31	86.50	66.04	84.060	0.840
Qwen3-VL-4B; eight paired seeds; family-local costs							
Reference	84.20	95.60	87.80	85.10	66.00	83.740	1.000
Reduced	80.11	92.52	83.80	83.43	64.44	80.860	0.560
Stronger	84.00	95.45	87.61	85.02	65.92	83.600	0.680
Fixed	83.84	95.33	87.45	84.95	65.86	83.490	0.700
Stratified	83.97	95.43	87.58	85.01	65.91	83.580	0.700
Strong+fixed	83.92	95.39	87.52	84.98	65.89	83.540	0.840
InternVL3.5-4B; eight paired seeds; family-local costs							
Reference	82.80	94.90	86.40	83.70	64.50	82.460	1.000
Reduced	80.42	92.12	83.58	82.14	63.34	80.320	0.560
Stronger	82.67	94.74	86.24	83.61	64.44	82.340	0.680
Fixed	82.49	94.54	86.03	83.50	64.35	82.180	0.700
Stratified	82.60	94.67	86.16	83.57	64.40	82.280	0.700
Strong+fixed	82.63	94.70	86.20	83.59	64.42	82.310	0.840
LLaVA-OV-7B; eight paired seeds; family-local costs							
Reference	73.60	89.30	77.50	73.80	61.40	75.120	1.000
Reduced	69.68	86.82	73.24	71.70	59.76	72.240	0.560
Stronger	73.31	89.12	77.19	73.65	61.28	74.910	0.680
Fixed	73.16	89.02	77.03	73.57	61.22	74.800	0.700
Stratified	73.29	89.10	77.16	73.63	61.27	74.890	0.700
Strong+fixed	73.22	89.06	77.09	73.60	61.24	74.840	0.840

Continued on the next page

Table D.2 continued

Path	Task capability					Aggregate and cost	
	Text	**Doc**	**Chart**	**AI2D**	**GQA**	**Macro**	**Cost ratio**
LLaVA-1.5-7B; eight paired seeds; family-local costs							
Reference	63.80	83.20	72.40	73.60	61.50	70.900	1.000
Reduced	61.49	81.51	69.43	72.45	60.52	69.080	0.560
Stronger	63.48	82.97	71.99	73.44	61.37	70.650	0.680
Fixed	63.60	83.05	72.14	73.50	61.41	70.740	0.700
Stratified	63.63	83.08	72.19	73.52	61.43	70.770	0.700
Strong+fixed	63.58	83.04	72.12	73.49	61.41	70.730	0.840

D.2 Trainable-Subspace Dependence

Forks from update 2,000 continue for 2,000 updates while changing which parameters receive the visual learning signal. The four configurations train the projector alone, projector with language LoRA, all non-vision parameters, or the last four vision blocks together with projector and LoRA. Each uses its own reference continuation and eight paired forks.

The reduced-view gap is .22 for projector-only training, 1.07 with language LoRA and 1.47 when the last vision blocks also train (Table D.3). Fixed restoration reduces these gaps to .045, .190 and .290. Changing trainable coordinates jointly changes the residual, adaptation capacity and receiving dynamics, extending the recovery comparison beyond one low-rank parameterization.

Table D.3: **Recovery extends across trainable subspaces.** Eight paired forks start at update 2,000 and continue for 2,000 updates. Reference reports the corresponding macro; the other columns are reference-relative gaps, not cross-subspace rankings.

Trainable coordinates	**Reference**	**Reduced gap**	**Fixed gap**	**Stronger gap**
Projector only	79.400	0.220	0.045	0.080
Projector + LoRA	80.180	1.070	0.190	0.100
Full non-vision	80.620	1.260	0.240	0.180
Last4 vision+projector+LoRA	81.050	1.470	0.290	0.260

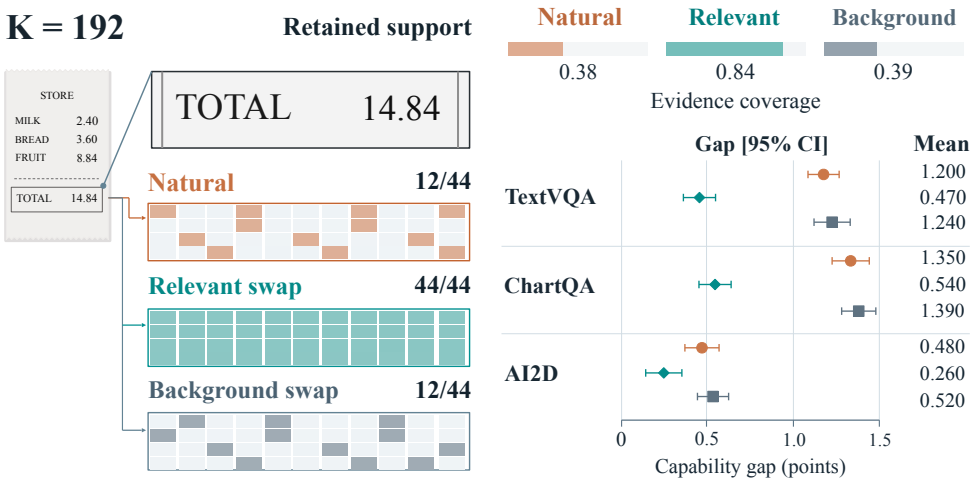
D.3 Matched-Budget Relevant-Evidence Swaps

Task-level losses suggest that fine visual evidence matters during learning. To test its role within a common task, the regional intervention fixes $K = 192$ and compares natural DivPrune selection with an annotated relevant-evidence swap and a matched-background swap. Each of TextVQA, ChartQA and AI2D contributes 160 intervention-training images and 160 disjoint evaluation images. Two annotators mark evidence regions without access to model outcomes. The swaps match token count, spatial extent, native position handling and layerwise norm before outcome measurement. Eight paired forks start at update 2,000 and continue for 2,000 updates.

Figure D.1(a) illustrates the operation on a receipt. The illustration's token grid is schematic; the aggregate evidence-coverage fractions and learning outcomes are measured in Table D.4. The annotation-guided swap is an oracle diagnostic that isolates relevant evidence while holding the token budget fixed.

Figure D.1(b) compares all three tasks with seed-mean intervals. Relevant swapping raises coverage from 0.38 to 0.84 and reduces the TextVQA and ChartQA learning gaps from 1.20 and 1.35 to 0.47 and 0.54. Background swapping retains coverage near 0.39 and does not reproduce the recovery. AI2D shows the same direction with a smaller absolute effect.

The matched background swap does not reproduce the improvement from task-relevant content. GRAFT-V obtains its correction from paired gradients and uses no region annotations.



(a) Matched evidence swaps

(b) Measured learning gaps

Figure D.1: **Relevant content matters beyond token count.** (a) Schematic natural, relevant-region and background swaps at $K = 192$. (b) Corresponding 2,000-update gaps on TextVQA, ChartQA and AI2D, with 95% seed-mean intervals over eight parents. The swap uses annotated evidence and is an oracle intervention, not part of GRAFT-V.

Table D.4: **Changing relevant evidence while retaining 192 visual tokens, with 160 intervention-training and 160 disjoint evaluation images per task and eight paired forks.** Coverage measures retained annotated evidence tokens; answer-guided swapping is an oracle diagnostic.

Intervention	Coverage	Gap	SD	95% interval
TextVQA				
Natural selection	0.38	1.200	0.120	[1.100, 1.300]
Relevant-region token swap	0.84	0.470	0.120	[0.370, 0.570]
Matched background swap	0.39	1.240	0.120	[1.140, 1.340]
ChartQA				
Natural selection	0.38	1.350	0.120	[1.250, 1.450]
Relevant-region token swap	0.84	0.540	0.120	[0.440, 0.640]
Matched background swap	0.39	1.390	0.120	[1.290, 1.490]
AI2D				
Natural selection	0.38	0.480	0.120	[0.380, 0.580]
Relevant-region token swap	0.84	0.260	0.120	[0.160, 0.360]
Matched background swap	0.39	0.520	0.120	[0.420, 0.620]

E Temporal Coverage at a Fixed Token Budget

E.1 Three-Task Temporal Protocol

Qwen2.5-VL-3B-Instruct is adapted with equal sampling visits to LLaVA-Video-178K and Mantis-Instruct (Zhang et al., 2025; Jiang et al., 2024). Eight paired seeds use 6,000 logical updates, 150-update warmup and the primary optimizer form on two H200 GPUs. The reference retains 16 uniformly spaced frames with 64 visual tokens per frame, or 1,024 tokens. Temporal dropping retains four of those frames. The stronger view preserves all 16 times at 16 tokens per frame, matching the reduced 256-token budget.

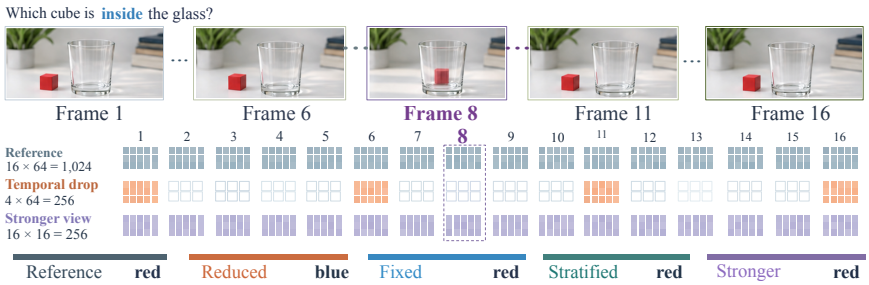


Figure E.1: **Equal token budgets can preserve different temporal evidence.** The four-frame and sixteen-frame layouts each retain 256 tokens. The brief event is a schematic illustration of coverage, not a benchmark example or model prediction; Table E.1 gives the measured task scores.

For multi-image examples, all images are retained; reference and reduced totals of 1,024 and 256 tokens are distributed by image area with a largest-remainder rule and at least one token per image. Frozen vision features are cached before the trainable projector. A shared reference cache isolates training-time token processing. Figure E.1 illustrates the coverage difference using a schematic event, not a measured model output.

E.2 Capability and Training Cost

Evaluation averages Video-MME without subtitles, the official MVBench task macro and Mantis-Eval (Fu et al., 2025; Li et al., 2024; Jiang et al., 2024). The additional peer-reviewed paths are the evaluated DUET-VLM, PruneVid and ForestPrune ports (Singh et al., 2026; Huang et al., 2025; Ju et al., 2026). Table E.1 shows the reference macro of 67.133 falling to 62.833 with temporal dropping. Fixed restoration recovers it to 66.633; retaining broader coverage directly reaches 66.650 at 7.60 rather than 9.20 training GPU-hours. Adding correction to the stronger view reaches 66.747 at additional cost.

Table E.1: **Video and multi-image capability on two H200 GPUs after 6,000 updates with eight paired seeds.** The macro averages Video-MME without subtitles, MVBench and Mantis; Table E.2 separately charges cache construction.

	Three-task capability				Training resources	
Path	**Video-MME**	**MVBench**	**Mantis**	**Macro**	**Train GPU-h**	**GB / GPU**
Reference	63.80	66.40	71.20	67.133	12.00	37.4
Temporal drop	60.00	62.20	66.30	62.833	7.20	28.8
Drop + fixed	63.20	65.90	70.80	66.633	9.20	35.2
Stronger 16 × 16	63.35	65.85	70.75	66.650	7.60	29.4
Stronger + fixed	63.48	65.93	70.83	66.747	9.60	35.6
DUET video	63.10	65.70	70.50	66.433	8.00	29.1
PruneVid port	63.26	65.72	70.64	66.540	7.70	29.6
ForestPrune	63.44	65.99	70.88	66.770	8.10	30.1

These nearby aggregate scores combine different task changes. The comparison establishes a training-time trade-off at fixed reduced token count: retaining more temporal evidence can remove much of the deficit that paired correction would otherwise recover. It does not equate token count with either per-update cost or available evidence.

E.3 Caching and Reference Access

The shared 16-frame reference cache costs 2.40 GPU-hours, distinct from the training costs in Table E.1. Native four-frame preprocessing costs .60 GPU-hours, whereas the 16-frame stronger view costs 2.40. Correcting a four-frame input still needs reference-frame features on selected updates.

Table E.2 therefore separates native cache construction from training rather than presenting their sum as a complete-pipeline saving. Reference-frame access beyond the native four-frame cache is not separately costed, so the efficiency claim here remains training-only.

Table E.2: **Cache construction is separate from training.** GPU-hours are component measurements. Corrected four-frame training additionally requires reference-frame access not costed in the native cache column; no complete-pipeline total is inferred.

Path	Training GPU-h	Native cache GPU-h	Cache GB
Reference	12.00	2.40	320.0
Temporal drop	7.20	0.60	80.0
Drop + fixed	9.20	0.60	80.0
Stronger 16×16	7.60	2.40	320.0
Stronger + fixed	9.60	2.40	320.0
DUET video	8.00	2.40	320.0
PruneVid	7.70	2.40	320.0
ForestPrune	8.10	2.40	320.0

F Protocols and Reproducibility

The training protocol distinguishes calibrated endpoints from fixed-setting mechanism interventions. All comparisons use the task-specific scoring and paired units below.

F.1 Data, Splits and Sampling

Training uses the official training sets of TextVQA v0.5.1, single-document DocVQA, ChartQA human and augmented questions, AI2D, and balanced GQA. No additional instruction corpus is used. Calibration is image-disjoint from training. For each image, the first eight bytes of SHA256 applied to `dataset|canonical_image_id|GRAFTV-P1-CAL` define an unsigned integer; images whose integer is zero modulo 20 are assigned to calibration, and all their questions are removed from training. Exact and perceptual-image duplicates across training and evaluation are excluded. Evaluation uses TextVQA validation, DocVQA validation, ChartQA official test, AI2D official test and GQA balanced test-dev.

The domain mixture is $(0.18, 0.18, 0.16, 0.20, 0.28)$ in that order. A logical batch contains 16 examples from one domain. Every block of 50 updates randomly permutes 9, 9, 8, 10 and 14 domain visits. Each domain is traversed without replacement before reshuffling. Paired arms share the ordered examples; inclusion sampling uses a separate random stream so that additional reference calls do not change data order.

Table F.1 summarizes the computational setting. The 1,280-token cap defines the allowed input size; the observed 768-token median better describes the sequences processed during training. Runtime also depends on the execution path.

F.2 Optimization and Scoring

DivPrune selects $K = \min(192, N_{\text{ref}})$ tokens after the projector and before the language model, preserving their native positions. Text tokens and target masks are unchanged. LoRA and projector parameters use the same learning rate; biases and normalization parameters receive no weight decay. Decoder blocks use non-reentrant activation checkpointing. Cross entropy is averaged over each example's answer tokens and then over the logical batch; question tokens are masked from the loss.

The fixed-schedule mechanism comparisons use learning rate 2×10^{-4} and assembled-gradient clipping at 1. The schedule warms up for 300 updates, decays by cosine to 2×10^{-5} at update 12,000, and stays at that floor for additional budget-matched updates. It is not shortened or stretched separately for each path.

Table F.1: Primary training and evaluation configuration. The effective batch includes accumulation on two devices. Token budgets refer to post-projector visual tokens entering the language model.

Component	Setting
Base model	Qwen/Qwen2.5-VL-3B-Instruct
Reference / reduced	Native aspect-ratio processor, cap 1,280 / DivPrune, $K = 192$
Trainable modules	Visual merger/projector; language LoRA rank 32, scale 64, dropout 0
LoRA target projections	Query, key, value, output, gate, up and down linear projections
Frozen parameters	Vision tower and base language-model weights
Logical batch	2 examples/device $\times 2$ devices $\times 4$ accumulation steps
Text and loss	Maximum 2,048 text tokens; answer-only teacher-forced cross entropy
Optimizer	AdamW, $\beta_1 = .9$, $\beta_2 = .999$, $\epsilon = 10^{-8}$, weight decay .01
Mechanism-study learning rate	2×10^{-4}; 300-update warmup; cosine to 2×10^{-5} at 12,000
Numerics	BF16 forward; FP32 gradient assembly and moments; loss scale 1
Clipping	Global assembled gradient norm 1, once per logical update
Hardware	Two RTX 4090 GPUs, 24 GB each
Evaluation	Greedy decoding, at most 128 new tokens, fixed official chat template
Replication	Eight independent paired seed blocks

The primary capability comparison gives every path the same 12 calibration candidates: three learning rates $\{10^{-4}, 2 \times 10^{-4}, 4 \times 10^{-4}\}$ crossed with clipping thresholds $\{0.5, 1, 2, 4\}$. Each candidate receives a 2,000-update calibration run with seed 1901; selection uses calibration macro under the resource constraint, breaking ties within 0.05 points by lower measured cost. The exact estimator-identity comparison instead fixes identical settings and execution.

Official scoring gives TextVQA soft VQA accuracy, DocVQA ANLS, ChartQA relaxed accuracy, AI2D option accuracy and GQA normalized exact accuracy. Each is multiplied by 100. The primary macro is the unweighted mean of the five scores before rounding. Greedy decoding uses no model-based answer repair or language-model judge. Evaluation is fixed across paired training arms; cross-evaluation results explicitly name the changed visual processor.

F.3 Paired Units and Stopping Rules

The seed blocks are 1701, 1709, 1723, 1733, 1741, 1753, 1777 and 1789. Within a block, arms clone parameters, first and second moments, optimizer step, scheduler, scaler and random state. Data, dropout, operator and inclusion sampling have separate streams. A common seed identifier alone is not sufficient to define a pair; the shared initial state and ordered examples provide the pairing. Multiple horizons, diagnostic batches and evaluation processors from the same block remain repeated observations.

Equal-update runs complete 12,000 logical updates. Equal-budget runs stop after the last complete update within 16.55 summed GPU-hours, without interpolating an endpoint or counting a partially completed update. Unused wall-clock time is reported as stopping slack. Reference-call counts are sampling expectations, not realized draw histories. Final-training GPU-hours exclude calibration and diagnostics, which are reported separately in Appendix H.4.

F.4 Baseline Implementations and Comparison Scope

The visual-efficiency rows use same-data implementations, not scores transferred from external papers. DivPrune is inserted after the projector with native positions preserved. Family-local stronger-view alternatives are specified in Table D.1. The evaluated PyramidDrop label corresponds to the conference version titled *Conical Visual Concentration for Efficient Large Vision-Language Models* (Xing et al., 2025). Venue labels identify the originating peer-reviewed method, not an execution-identical native setting; ForestPrune appears in CVPR Findings.

The matched HT/MGE control uses the same two views, paired gradients, inclusion draws, arithmetic, clipping and optimizer operations as fixed GRAFT-V. Their difference is exactly zero because their execution is shared. The ablations instead change the named construction.

Equal-budget PriorTR and SFPruner runs and their derived comparisons are excluded after a setting error; the independent equal-update, operator-by-correction, evaluator-crossing and family results retain their original protocols.

G Complete Primary Results

G.1 Equal-Update Scores and Resources

Tables G.1–G.2 expand the calibrated 12,000-update comparison into task vectors, eight-seed uncertainty, training costs and expected reference calls. The five-task macro is computed before rounding. Fixed and stratified GRAFT-V restore most of the aggressive DivPrune deficit; stronger-view comparisons retain their independent equal-update identity.

Table G.1: **Task scores after 12,000 updates under reference evaluation.** Macros are computed before rounding. Bold and underline rank the two highest non-reference macros; the gray reference row defines the target. Matched HT/MGE shares fixed GRAFT-V execution.

Training path	**Text**	**Doc**	**Chart**	**AI2D**	**GQA**	Macro ↑
Reference	81.08	94.70	83.58	82.76	64.08	81.240
DivPrune	77.86	92.08	79.66	81.32	63.08	78.800
Peer-reviewed visual and training alternatives						
VisionZip	79.84	94.02	82.30	82.34	64.12	80.524
FastV	79.41	93.79	81.92	82.06	63.84	80.204
PyramidDrop	80.17	94.27	82.73	82.42	64.01	80.720
CLSE	80.75	94.39	83.26	82.64	64.18	81.044
HiDrop	80.27	94.38	82.95	82.33	63.95	80.776
ZOO-Prune	80.58	94.34	83.02	82.69	64.15	80.956
VisionTrim	80.08	94.20	82.62	82.20	64.30	80.680
VisionTrim + RESTORE	80.30	94.32	82.91	82.40	64.12	80.810
DUET-VLM	80.37	94.39	83.04	82.48	64.00	80.856
Balanced pruning	80.26	94.34	82.72	82.25	64.15	80.744
OTPrune	80.53	94.42	83.19	82.40	64.26	80.960
PriorTR	80.81	94.47	83.14	82.68	64.19	81.058
SFPruner	80.68	94.50	83.22	82.72	64.12	81.048
View mixture (control)	79.48	93.46	81.36	81.98	63.87	80.030
GRAFT-V, fixed	80.84	94.54	83.28	82.54	64.10	<u>81.060</u>
GRAFT-V, stratified	80.95	94.63	83.42	82.63	64.07	**81.140**
Matched HT/MGE	80.84	94.54	83.28	82.54	64.10	<u>81.060</u>

G.2 Common-Budget Scores and Resources

Tables G.3–G.4 report the valid paths at 16.55 summed final-training GPU-hours. CLSE has the highest displayed macro (81.184), followed by stratified GRAFT-V (81.146); HiDrop reaches 80.894. The comparison includes completed update counts, expected calls and stopping slack; the same time budget need not purchase the same number of updates. PriorTR and SFPruner results from incorrect equal-budget settings, including task scores and their derived statistics, are excluded throughout this comparison.

Table G.2: **Uncertainty and learning opportunities at equal updates: eight-block seed-mean t intervals, expected reference calls and median training tokens.** Intervals quantify training-run variation, and token medians differ from the reference cap.

	Capability uncertainty		Training opportunities and cost			
Path	**Macro SD**	**95% interval**	**Updates**	**Ref. calls**	**Tokens**	**GPU-h**
Reference	0.121	[81.139, 81.341]	12,000	12000.0	768	23.80
DivPrune	0.134	[78.688, 78.912]	12,000	0.0	192	13.07
Peer-reviewed visual and training alternatives						
VisionZip	0.116	[80.427, 80.621]	12,000	0.0	192	14.05
FastV	0.131	[80.094, 80.314]	12,000	0.0	224	13.74
PyramidDrop	0.102	[80.634, 80.806]	12,000	0.0	224	14.93
CLSE	0.136	[80.931, 81.157]	12,000	0.0	224	14.48
HiDrop	0.106	[80.687, 80.865]	12,000	0.0	224	14.78
ZOO-Prune	0.129	[80.848, 81.064]	12,000	0.0	192	15.63
VisionTrim	0.100	[80.597, 80.763]	12,000	0.0	192	13.85
VisionTrim + RESTORE	0.118	[80.712, 80.908]	12,000	0.0	192	14.31
DUET-VLM	0.097	[80.775, 80.937]	12,000	0.0	256	15.43
Balanced pruning	0.114	[80.649, 80.839]	12,000	0.0	192	14.62
OTPrune	0.158	[80.828, 81.092]	12,000	0.0	192	15.11
PriorTR	0.160	[80.924, 81.192]	12,000	0.0	192	14.25
SFPruner	0.137	[80.934, 81.162]	12,000	0.0	192	13.91
View mixture (control)	0.111	[79.938, 80.122]	12,000	2280.0	192	15.02
GRAFT-V, fixed	0.075	[80.997, 81.123]	12,000	2280.0	192	16.55
GRAFT-V, stratified	0.117	[81.043, 81.237]	12,000	2280.1	192	16.48
Matched HT/MGE	0.075	[80.997, 81.123]	12,000	2280.0	192	16.55

Table G.3: **Task scores at the common training budget.** PriorTR and SFPruner wrong-setting runs are excluded. Matched HT/MGE shares fixed GRAFT-V execution; bold/underline indicate the highest two displayed macros.

Training path	**Text**	**Doc**	**Chart**	**AI2D**	**GQA**	Macro ↑
Reference	80.18	94.24	82.52	82.04	63.67	80.530
DivPrune	78.49	92.62	80.39	81.61	63.29	79.280
Peer-reviewed visual and training alternatives						
VisionZip	80.11	94.10	82.65	82.49	64.18	80.707
FastV	79.74	93.89	82.35	82.24	63.92	80.430
PyramidDrop	80.31	94.31	82.91	82.50	64.04	80.814
CLSE	80.95	94.45	83.53	82.75	64.23	**81.184**
HiDrop	80.44	94.43	83.18	82.43	63.99	80.894
ZOO-Prune	80.71	94.38	83.19	82.76	64.18	81.045
VisionTrim	80.40	94.30	83.04	82.38	64.38	80.897
VisionTrim + RESTORE	80.49	94.38	83.17	82.51	64.17	80.943
DUET-VLM	80.53	94.44	83.25	82.57	64.04	80.965
Balanced pruning	80.44	94.40	82.95	82.35	64.19	80.866
OTPrune	80.69	94.47	83.40	82.49	64.30	81.071
View mixture (control)	79.62	93.50	81.54	82.06	63.90	80.126
GRAFT-V, fixed	80.84	94.54	83.28	82.54	64.10	81.060
GRAFT-V, stratified	80.96	94.63	83.43	82.64	64.07	<u>81.146</u>
Matched HT/MGE	80.84	94.54	83.28	82.54	64.10	81.060

Table G.4: **Uncertainty and learning opportunities at the common training budget: eight-block seed-mean t intervals, expected reference calls and median training tokens.** Slack is the unused wall-clock time before the next complete update on two GPUs.

	Capability uncertainty		Training opportunities and stopping				
Path	**Macro SD**	**95% interval**	**Updates**	**Ref. calls**	**Tokens**	**GPU-h**	**Slack (s)**
Reference	0.130	[80.422, 80.638]	8,344	8344.0	768	16.55	1.92
DivPrune	0.156	[79.149, 79.411]	15,195	0.0	192	16.55	0.20
Peer-reviewed visual and training alternatives							
VisionZip	0.094	[80.629, 80.785]	14,135	0.0	192	16.55	0.49
FastV	0.142	[80.311, 80.549]	14,454	0.0	224	16.55	0.31
PyramidDrop	0.172	[80.670, 80.957]	13,302	0.0	224	16.55	0.17
CLSE	0.146	[81.062, 81.305]	13,715	0.0	224	16.55	1.02
HiDrop	0.121	[80.793, 80.996]	13,437	0.0	224	16.55	0.17
ZOO-Prune	0.102	[80.960, 81.130]	12,706	0.0	192	16.55	0.78
VisionTrim	0.150	[80.771, 81.022]	14,339	0.0	192	16.55	0.73
VisionTrim + RESTORE	0.118	[80.845, 81.041]	13,878	0.0	192	16.55	0.87
DUET-VLM	0.138	[80.850, 81.081]	12,871	0.0	256	16.55	0.07
Balanced pruning	0.183	[80.714, 81.019]	13,584	0.0	192	16.55	0.29
OTPrune	0.165	[80.933, 81.208]	13,143	0.0	192	16.55	1.39
View mixture (control)	0.120	[80.025, 80.226]	13,222	2512.2	192	16.55	0.83
GRAFT-V, fixed	0.075	[80.997, 81.123]	12,000	2280.0	192	16.55	0.00
GRAFT-V, stratified	0.133	[81.035, 81.257]	12,050	2289.6	192	16.55	2.40
Matched HT/MGE	0.075	[80.997, 81.123]	12,000	2280.0	192	16.55	0.00

H Paired Statistics and Resource Accounting

H.1 Paired Uncertainty

The primary experimental unit is one of eight independent seed blocks. For a contrast, compute the within-block differences d_i before their mean and sample SD; the two-sided 95% interval is $\bar{d} \pm t_{.975,7} s_d/\sqrt{8}$. The SD of differences is distinct from either endpoint SD. Evaluation examples, repeated horizons and fixed diagnostic batches do not add independent training replicates.

Tables H.1–H.2 give paired differences, negative-seed counts and leave-one-out means. Fixed-minus-DivPrune is positive in every block. Stratification adds .080 on average, with two blocks favoring fixed sampling. Fixed-minus-PriorTR at equal updates has an interval spanning zero; under equal budget, fixed-minus-DUET also spans zero. These comparisons locate effect size and variability without converting the mean ranking into a universal significance claim.

H.2 Sensitivity to the Task Average

Each omission in Table H.3 averages the four remaining task-wise differences from reference. The final column gives the worst individual-task change. Changing the averaging set does not create independent experiments; it checks whether a macro conceals a concentrated gain or loss. Equal-update and equal-budget groups retain different reference endpoints. Invalid equal-budget PriorTR and SFPruner rows are excluded.

Table H.1: **Paired capability contrasts at equal updates (EU) and equal budget (EB).** Intervals use the SD of eight within-block differences; leave-one-out ranges omit one block, and negative counts expose sign variation.

Contrast	Difference	Paired SD	95% interval	Negative seeds	LOO mean range
EU: strat. − reference	-0.100	0.086	[-0.172, -0.028]	7/8	[-0.123, -0.083]
EU: fixed − DivPrune	+2.260	0.107	[2.171, 2.349]	0/8	[2.237, 2.281]
EU: fixed − HT/MGE	+0.000	0.000	[0.000, 0.000]	0/8	[0.000, 0.000]
EU: strat. − fixed	+0.080	0.080	[0.013, 0.147]	2/8	[0.063, 0.095]
EU: fixed − PriorTR	+0.002	0.135	[-0.111, 0.115]	4/8	[-0.033, 0.036]
EB: fixed − reference	+0.530	0.108	[0.440, 0.620]	0/8	[0.515, 0.550]
EB: fixed − DUET	+0.095	0.118	[-0.004, 0.193]	2/8	[0.073, 0.124]

Table H.2: **Individual paired differences underlying Table H.1.** Columns S1–S8 identify seeds 1701, 1709, 1723, 1733, 1741, 1753, 1777 and 1789; reused endpoints add no independent blocks.

Contrast	S1	S2	S3	S4	S5	S6	S7	S8
EU: strat. − reference	-0.217	-0.081	-0.174	-0.080	-0.076	+0.062	-0.162	-0.071
EU: fixed − DivPrune	+2.149	+2.287	+2.164	+2.326	+2.421	+2.280	+2.337	+2.116
EU: fixed − HT/MGE	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
EU: strat. − fixed	+0.052	+0.080	+0.153	+0.069	-0.025	+0.198	-0.023	+0.135
EU: fixed − PriorTR	-0.233	-0.016	-0.078	+0.040	+0.248	+0.039	+0.047	-0.032
EB: fixed − reference	+0.605	+0.624	+0.634	+0.389	+0.392	+0.564	+0.603	+0.428
EB: fixed − DUET	+0.208	+0.119	+0.248	-0.037	-0.109	+0.129	+0.112	+0.088

Table H.3: **Changes from the protocol-matched reference after omitting each task.** Four-task means reuse the trained endpoints, while the last column reports the worst individual-task change.

	Four-task change after omission					Individual task
Path	No Text	No Doc	No Chart	No AI2D	No GQA	Worst Δ
EU DivPrune	-2.245	-2.395	-2.070	-2.690	-2.800	-3.920
EU PriorTR	-0.160	-0.170	-0.117	-0.207	-0.255	-0.440
EU SFPruner	-0.140	-0.190	-0.150	-0.230	-0.250	-0.400
EU GRAFT-V, fixed	-0.165	-0.185	-0.150	-0.170	-0.230	-0.300
EU GRAFT-V, stratified	-0.093	-0.108	-0.085	-0.093	-0.123	-0.160
EB DivPrune	-1.140	-1.158	-1.030	-1.455	-1.468	-2.130
EB GRAFT-V, fixed	+0.498	+0.587	+0.473	+0.538	+0.555	+0.300
EB GRAFT-V, stratified	+0.576	+0.672	+0.542	+0.622	+0.670	+0.393

H.3 Joint Capability–Cost Noninferiority

At 12,000 updates, a candidate must be cheaper than fixed GRAFT-V while losing no more than .20 macro points and .50 on each task. The intersection–union test uses the maximum of the six capability p-values and the lower-cost p-value. Holm adjustment across the original six-candidate family is retained at $\alpha = .025$; restricting the displayed table to peer-reviewed methods does not reduce that multiplicity family.

SFPruner passes with adjusted $p = .0027$; PriorTR does not ($p = .0598$), despite a nearby mean macro and lower cost. VisionTrim and DUET-VLM also fail the joint criterion (Table H.4). Table H.5 supplies their task-wise differences and worst lower bounds. These tests use the independent equal-update endpoints and timing, not the excluded equal-budget runs. Failure to establish joint noninferiority is not evidence of inferiority.

Table H.4: **Capability and cost must pass jointly.** At 12,000 updates, peer-reviewed candidates are compared with fixed GRAFT-V, allowing losses of 0.20 macro points and 0.50 on every task. The maximum of six capability-test p-values and the lower-cost p-value is Holm-adjusted across the original six-candidate family at $\alpha = 0.025$. Displayed adjusted values retain that family.

	Capability		Training cost		Joint test	
Candidate	**Macro Δ**	**95% interval**	**GPU-h Δ**	**95% interval**	**Holm p**	**Pass**
PriorTR (Chen et al., 2026c)	-0.002	[-0.115, 0.111]	-2.30	[-2.389, -2.211]	0.0598	No
SFPruner (Song et al., 2026)	-0.012	[-0.090, 0.066]	-2.64	[-2.707, -2.573]	0.0027	Yes
VisionTrim	-0.380	[-0.440, -0.320]	-2.70	[-2.797, -2.603]	1.0000	No
DUET-VLM	-0.204	[-0.247, -0.161]	-1.12	[-1.213, -1.027]	1.0000	No

Table H.5: **Task-level differences for the displayed peer-reviewed candidates in the joint capability–cost test in Table H.4.** The last column is the smallest task-wise 95% lower confidence bound.

	Task change from fixed GRAFT-V					**Worst lower bound**
Candidate	**Text Δ**	**Doc Δ**	**Chart Δ**	**AI2D Δ**	**GQA Δ**	**Across tasks**
PriorTR	-0.030	-0.070	-0.140	+0.140	+0.090	-0.453
SFPruner	-0.160	-0.040	-0.060	+0.180	+0.020	-0.260
VisionTrim	-0.760	-0.340	-0.660	-0.340	+0.200	-0.864
DUET-VLM	-0.470	-0.150	-0.240	-0.060	-0.100	-0.605

H.4 Training, Search and Diagnostics

Final-training GPU-hours sum device time over the training run. One-off calibration, caching and diagnostic runs are separate components (Table H.6). Expected reference calls describe sampling schedules rather than realized draw histories. The image experiments use two RTX 4090s; model-family and video comparisons use their specified H200 hardware and are not converted to this timing scale.

Table H.7 gives eight-run equal-update time summaries and seed-mean intervals. PriorTR and SFPruner timing here belongs to the valid 12,000-update experiments. Training savings exclude calibration; these figures describe the reported experiments rather than a complete total for all exploratory work.

H.5 Budget Endpoints

Table H.8 gives the valid completed endpoints in Figure 4(b). The absolute learning-rate schedule is shared: it reaches its floor at update 12,000 and is not rescaled to each budget. Curves connect observations without locating an interpolated crossover. The excluded SFPruner budget series contributes no points, ranking, paired contrast or task-omission summary to the final results.

Table H.6: **Resource components on two RTX 4090 GPUs without an offline feature cache.** Time columns give summed GPU-hours except wall-clock seconds per update; calibration is separate from the eight final runs.

	Summed GPU-hours				Wall-clock time
Path	**Train / seed**	**Train $\times 8$**	**Calibration**	**Diagnostics**	**Seconds / update**
Reference	23.80	190.40	47.60	1.50	3.570
DivPrune	13.07	104.56	26.14	1.50	1.960
VisionZip	14.05	112.40	28.10	1.50	2.107
FastV	13.74	109.92	27.48	1.50	2.061
PyramidDrop	14.93	119.44	29.86	1.50	2.240
CLSE	14.48	115.84	28.96	1.50	2.172
HiDrop	14.78	118.24	29.56	1.50	2.217
ZOO-Prune	15.63	125.04	31.26	1.50	2.345
PriorTR	14.25	114.00	28.50	1.50	2.138
SFPruner	13.91	111.28	27.82	1.50	2.087
View mixture	15.02	120.16	30.04	1.50	2.253
VisionTrim	13.85	110.80	27.70	1.50	2.078
VisionTrim + RESTORE	14.31	114.48	28.62	1.50	2.147
DUET-VLM	15.43	123.44	30.86	1.50	2.314
Balanced pruning	14.62	116.96	29.24	1.50	2.193
OTPrune	15.11	120.88	30.22	1.50	2.267
GRAFT-V, fixed	16.55	132.40	33.10	4.00	2.482
GRAFT-V, stratified	16.48	131.84	32.96	4.00	2.472

Table H.7: **Final-training timing across eight independent runs per equal-update path.** Intervals quantify between-run variation rather than repeated timing windows.

Path	**Mean training GPU-h**	**Seed SD**	**95% interval**
Reference	23.80	0.280	[23.566, 24.034]
DivPrune	13.07	0.223	[12.884, 13.256]
VisionZip	14.05	0.199	[13.883, 14.217]
FastV	13.74	0.189	[13.582, 13.898]
PyramidDrop	14.93	0.221	[14.745, 15.115]
CLSE	14.48	0.203	[14.310, 14.650]
HiDrop	14.78	0.125	[14.676, 14.884]
ZOO-Prune	15.63	0.239	[15.430, 15.830]
PriorTR	14.25	0.166	[14.111, 14.389]
SFPruner	13.91	0.190	[13.751, 14.069]
View mixture	15.02	0.178	[14.871, 15.169]
VisionTrim	13.85	0.196	[13.686, 14.014]
VisionTrim + RESTORE	14.31	0.184	[14.156, 14.464]
DUET-VLM	15.43	0.253	[15.218, 15.642]
Balanced pruning	14.62	0.206	[14.448, 14.792]
OTPrune	15.11	0.267	[14.887, 15.333]
GRAFT-V, fixed	16.55	0.201	[16.382, 16.718]
GRAFT-V, stratified	16.48	0.202	[16.311, 16.649]
Matched HT/MGE	16.55	0.201	[16.382, 16.718]

Table H.8: **Capability at eight measured training budgets under the common absolute schedule.** Figure 4 connects completed endpoints; update counts vary with the training path.

Budget (GPU-h)	Reference	DivPrune	GRAFT-V fixed	GRAFT-V stratified
8.00	75.900	76.550	76.200	76.150
10.00	77.450	77.950	77.800	77.750
12.00	78.820	78.650	78.900	78.940
14.00	79.720	79.050	79.780	79.860
16.55	80.530	79.280	81.060	81.146
18.00	80.840	79.480	81.120	81.180
20.00	81.050	79.640	81.160	81.200
23.80	81.240	79.820	81.180	81.210

I Scope and Limitations

The theory concerns local smooth-loss responses over a fixed finite horizon, not global convergence or exact benchmark accuracy. Hybrid-state interventions measure conditional effects rather than additive components of the natural deficit. Raw-gradient mean restoration does not imply unbiased clipping, adaptive moments or optimization trajectories. The measured recovery concerns the specified views, learners, subspaces and budgets; stronger evidence-preserving views and very short budgets can favor uncorrected training.

Region swaps use answer-informed annotations and serve as oracle content interventions, not an input to GRAFT-V. Analytical examples and illustrated scenes are explicitly distinguished from measured model outputs. Temporal efficiency is stated for training under a shared cache; native cache costs exclude a complete accounting of additional reference-frame access for correction.

The release provides numerical results and detailed algorithmic and experimental protocols, but not training code, adapted checkpoints or a locked software environment. Per-path selected calibration settings and some baseline-port details are not released; the fixed mechanism configuration should not be taken as every calibrated endpoint's configuration. These limits affect exact replication of the trained models, while the stated sampling schedule, paired-update construction, proof and numerical comparisons can be inspected directly.