WHEN DOES LOCAL REFINEMENT HELP? DIAGNOSING ROUTING TARGETS FOR SPARSE VIDEO FRAME INTERPOLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse video frame interpolation allocates a local Expert to a small subset of image regions. Effective allocation requires distinguishing prediction difficulty from the improvement that the available Expert can deliver. We freeze a full-frame predictor and a local residual Expert, then measure Actual Expert Gain: the signed reduction in reconstruction error from each tile-level intervention. An additive formulation separates the target-aware oracle, the feature-conditioned optimum, and the learned policy. On 500 Vimeo90K development examples, 37.59% of 56,000 interventions increase error. At six selected tiles per image, the gain oracle improves PSNR by 0.243 dB, compared with 0.132 dB for endpoint-disagreement routing. On a pre-frozen locked evaluation of 3,281 valid examples, three GainGate seeds show no statistically resolved improvement over Uncertainty at six or twelve selected tiles (paired image-bootstrap 95% confidence intervals include zero). Separately, vectorizing identical halo crops yields a 1.62× local-stage median speedup, while full sparse and dense systems have near-matched CPU latency. These results distinguish realized intervention value, its predictability from deployable features, and its execution cost, which must be evaluated separately when assessing sparse refinement.

Anonymous code:
`https://anonymous.4open.science/r/local-refinement-routing-EBD6`

1 INTRODUCTION

Video frame interpolation (VFI) reconstructs an intermediate frame from two temporal endpoints. Modern systems combine correspondence estimation with image synthesis, and efficient variants reduce the cost of both operations (Jiang et al., 2018; Bao et al., 2019; Huang et al., 2022; Kong et al., 2022; Li et al., 2023). A complementary strategy predicts the full frame with a lightweight path and spends additional computation only on selected regions. Coarse-to-fine refinement, motion-adaptive inference, uncertainty-guided pruning, and spatial selective refinement all make such allocation relevant to VFI (Yuan et al., 2019; Choi et al., 2021; Cheng et al., 2023; Hu et al., 2022; 2024). Figure 1(a) shows a real development example in which the local Expert reduces tile error. The operative systems question is whether the same Expert would help on the other candidate tiles competing for a limited number of calls.

The inexpensive signals examined here describe quantities other than the benefit of the deployed Expert. Motion measures endpoint change; *Uncertainty*, our experimental name for a coarse-to-endpoint disagreement score, measures discrepancy rather than calibrated predictive uncertainty; Teacher Advantage measures the improvement of a different model. A local Expert need not reproduce the teacher's correction and can worsen an adequate coarse prediction. Figure 1(b) shows both effects of limited repairability: 37.59% of its 56,000 candidate interventions increase error, while target-aware gain ranking identifies substantially more useful selections than the input-only router at the same six-tile budget. The question is which calls this Expert can usefully perform, not which regions look difficult in isolation.

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.

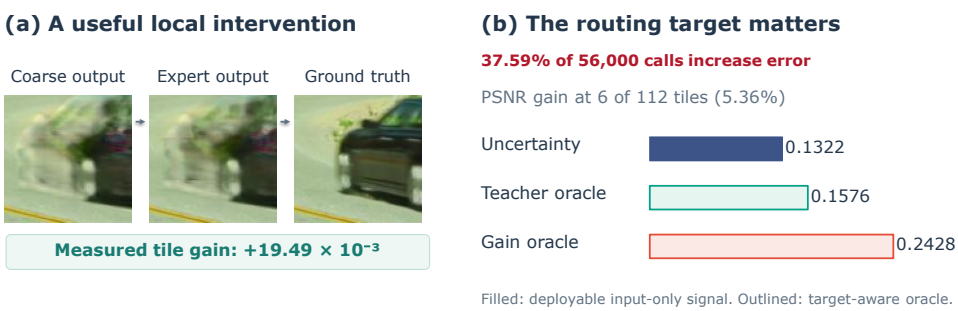


Figure 1: Local repairability and routing headroom. (a) A positive-gain development example; gain is the reduction in tile MSE. (b) Of 56,000 candidate interventions across 500 images, 37.59% increase error. At six of 112 tiles, input-only endpoint disagreement yields 0.1322 dB, while target-aware Teacher and Gain oracles yield 0.1576 and 0.2428 dB using the same local Expert. The example is selected by gain; aggregate statistics use all development images.

We study the resulting decision problem under a fixed local Expert, fixed crop geometry, and an exact integer budget. For each candidate tile, *Actual Expert Gain* is the signed reduction in reconstruction error produced by executing that same Expert. This definition turns every candidate into a measured intervention, including calls that worsen the coarse output. It also connects spatial routing to learning to defer, expert cascades, and decision-focused prediction, where the appropriate decision depends on the capability of the downstream action (Mozannar and Sontag, 2020; Jitkrittum et al., 2023; Elmachtoub and Grigas, 2022). Our question is then precise: how much intervention value exists, and how much can a Gate recover from information available before the target frame is observed?

Two challenges separate those quantities. First, the target-aware oracle observes every intervention outcome, whereas a deployable Gate receives only inexpensive input-derived features. An oracle margin can therefore combine value hidden from those features with error from approximation, finite data, or optimization. Second, fewer local operations do not automatically imply lower latency. Dispatch, memory access, crop handling, batching, and the host platform all affect measured execution (Ma et al., 2018; Narasimhan et al., 2025). Without controlled comparisons, a quality difference can be attributed to routing even when the coarse predictor changed, and an execution difference can be attributed to sparsity even when only the schedule changed.

Our analysis fixes the Light Path, local Expert, crop geometry, and tile budget across routing policies. A tile Gate reuses early Light-Path features together with endpoint motion and disagreement cues. Selected tiles receive halo context, run as one Expert batch, and write residuals only to disjoint interiors. The additive MSE objective makes the top-k signed-gain oracle exact for this fixed intervention, while an information-conditioned decomposition separates full-information headroom, the best value available from Gate features, and learned-policy performance. We test the same Light–Expert pair on development data and a pre-frozen set of 3,281 valid examples. For this locked-set comparison, confidence intervals are computed on the three-seed mean and reflect paired image-sampling uncertainty, not seed uncertainty. Output-matched vectorization and a halo ablation separate scheduling from context, while a near-latency-matched dense reference characterizes the complete systems.

Our contributions are threefold. First, we give a fixed-intervention formulation in which disjoint tile writes make signed-gain top-k selection exact for frame MSE; its information-conditioned analysis explains why target-aware headroom is not itself a measure of learnable improvement (Section 3). Second, a controlled study quantifies this distinction: target-aware selection exposes unused local refinement value, but the tested gain-supervised Gates remain close to endpoint disagreement in the reported mean-PSNR comparisons. Resized-domain transfer and Expert perturbations examine how these observations vary with the budget, domain, and available action (Section 5). Third, output-matched execution and halo controls distinguish batching gains from changes in reconstruction, while a measured dense reference contextualizes end-to-end cost. Together, the formulation

and controlled comparisons characterize allocation quality and execution cost for fixed-Expert refinement.

2 Related Work

Video frame interpolation. Existing VFI methods improve interpolation through kernel-based synthesis, optical-flow estimation, feature refinement, or attention-based reconstruction (Niklaus et al., 2017; Jiang et al., 2018; Bao et al., 2019; Choi et al., 2020; Xue et al., 2019). Coarse-to-fine instance refinement and many-to-many splatting provide two relevant precedents for allocating or refining spatial evidence (Yuan et al., 2019; Hu et al., 2022). Efficient models such as RIFE, IFRNet, and AMT emphasize the quality–runtime trade-off (Huang et al., 2022; Kong et al., 2022; Li et al., 2023). Recent work also studies interactive control, diffusion-based trajectories, large displacement, and ambiguous motion (Wang et al., 2025; Ma et al., 2026; Liu et al., 2024; Zhong et al., 2024). Our focus is not a new correspondence representation, but the decision of where an already chosen local Expert should act.

Conditional computation. DynamicViT removes redundant tokens, TokenLearner compresses dense features into learned tokens, and AdaFocus allocates video-recognition computation to selected regions (Rao et al., 2021; Ryoo et al., 2021; Wang et al., 2022). Mixture-of-Experts research studies assignment and router–expert coupling (Puigcerver et al., 2024; Lv et al., 2026); related conditional-computation work studies dynamic layer routing (Heakl et al., 2026). Dynamic VFI systems adapt computation using motion or uncertainty. MADA selects inference paths according to motion characteristics (Choi et al., 2021); uncertainty-guided pruning removes low-value spatial work (Cheng et al., 2023); and M2M++ extends many-to-many splatting with spatial selective refinement (Hu et al., 2022; 2024). These methods motivate spatial allocation. We isolate a different variable: whether one fixed local operation improves each candidate region.

Routing targets. Learning to defer and decision-focused prediction align decisions with downstream outcomes (Mozannar and Sontag, 2020; Elmachtoub and Grigas, 2022). Most directly, Jitkrittum et al. (2023) regress signed correctness or class-probability differences between frozen classifiers and show that a post-hoc router can overfit. We instantiate this outcome-based perspective for spatial reconstruction: a fixed cropped Expert writes disjoint residual interiors, continuous signed MSE reductions determine an exact-cardinality oracle, and runtime is measured independently of allocation quality. Speculative cascades likewise distinguish a deferral rule from its execution (Narasimhan et al., 2025), while unified routing and cascading identify quality estimation as central to model selection (Dekoninck et al., 2025). Our empirical focus is the magnitude of useful and harmful local interventions and the performance of inexpensive predictors of those interventions across the reported budgets and domains.

3 Fixed-Expert Routing

3.1 Prediction, action, and budget

Let $I_0, I_1 \in [0,1]^{3\times H\times W}$ be temporal endpoints and I_{gt} the intermediate target. A lightweight full-frame predictor produces I_c. Partition the image into N equal-area, non-overlapping output tiles $\mathcal{T}$. A Gate assigns scores s_t and selects an exact integer budget $k = \lceil bN \rceil$:

$$\mathcal{S} = \mathrm{TopK}(\{s_t : t \in \mathcal{T}\}, k), \qquad \hat{I} = \mathrm{clip}(I_c + M_{\mathcal{S}} \odot R_E, 0, 1). \tag{1}$$

Here $M_{\mathcal{S}}$ marks selected interiors and R_E contains their Expert residuals. The Light Path and Expert remain fixed throughout routing comparisons. Target-aware oracles alter the selected set but still execute the same local Expert; selecting by Teacher Advantage does not replace the output with the teacher prediction.

3.2 Shared features and vectorized local execution

The Light Path is a compact U-Net-like interpolator with channel widths $\{16, 32, 64\}$. It predicts a residual over the temporal average:

$$I_c = \mathrm{clip}\left((I_0 + I_1)/2 + f_L([I_0, I_1]), 0, 1\right). \tag{2}$$

The Gate reuses its first encoder feature F, pooled to the tile grid and detached from the Light Path. Additional cues are endpoint motion m and coarse-to-endpoint disagreement u:

$$m(x) = \frac{1}{3}\sum_c |I_0^c(x) - I_1^c(x)|, \tag{3}$$

$$u(x) = \frac{1}{6}\sum_c \big(|I_c^c(x) - I_0^c(x)| + |I_c^c(x) - I_1^c(x)|\big). \tag{4}$$

Three 1×1 convolutions produce tile scores. We retain the experimental name *Uncertainty* for the disagreement heuristic u. For each channel, $|c-a| + |c-b| = |a-b| + 2\,\mathrm{dist}(c, [\min(a,b), \max(a,b)])$. Thus $u = m/2$ when every coarse channel lies between its endpoint values. The two baselines can therefore induce the same ranking without measuring Expert competence; the general identity appears in Appendix A.

The Expert receives $[I_0, I_1, I_c]$ and predicts a residual. Each 32×32 output tile has an eight-pixel halo, giving a 48×48 input crop. The halo supplies context; only the interior is scattered back. Let P_t extract a halo crop and C remove its halo. Vectorized refinement is

$$X_{\mathcal{S}} = \mathrm{Stack}_{t\in\mathcal{S}}\, P_t([I_0, I_1, I_c]), \qquad R_E = \mathrm{Scatter}(C(f_E(X_{\mathcal{S}})), \mathcal{S}). \tag{5}$$

The looped implementation invokes f_E once per tile; the vectorized version invokes it once per selected batch. Input halos may overlap while output interiors remain disjoint, allowing context sharing without overlapping residual writes. Runtime includes indexing and composition.

3.3 Signed gain and the oracle objective

For tile t, let $I_{e,t}$ be its clipped prediction after the frozen Expert call. Define

$$G_t = \mathrm{MSE}(I_{c,t}, I_{gt,t}) - \mathrm{MSE}(I_{e,t}, I_{gt,t}). \tag{6}$$

Positive gain identifies a beneficial call; negative gain identifies a harmful one. A Teacher-Advantage score evaluates the corresponding teacher prediction and therefore a different intervention. Gain is specific to the frozen models and crop geometry; changing the available action changes the target.

Proposition 1 (Exact-budget allocation). For equal-area, disjoint output tiles, suppose each Expert output is independent of which other tiles are evaluated in the same batch. Then frame MSE satisfies

$$E(\mathcal{S}) = E_c - \frac{1}{N}\sum_{t\in\mathcal{S}} G_t. \tag{7}$$

Consequently, the k largest signed gains minimize frame MSE at $|\mathcal{S}| = k$. The claim follows by replacing coarse error with Expert error on selected interiors. It also maximizes each frame's PSNR because PSNR decreases monotonically with positive MSE. Additivity is specific to MSE; SSIM and LPIPS are evaluated separately. Under an at-most-k budget, the oracle selects only positive gains. Under an exact budget, it may include non-positive calls. Zero clipping preserves the positive prefix but erases ordering in the non-positive tail; absolute-value ranking can prioritize strongly harmful calls.

3.4 Oracle headroom and information available to the Gate

A target-aware oracle differs from the best decision available to an input-only Gate. Let Z collect the Gate's observable features and let $\mu_t(Z) = \mathbb{E}[G_t \mid Z]$. Define expected selected gain

$$\begin{aligned} V_{\mathrm{or}} &= \mathbb{E}\max_{|\mathcal{S}|=k}\sum_{t\in\mathcal{S}} G_t, \quad V_Z = \mathbb{E}\max_{|\mathcal{S}|=k}\sum_{t\in\mathcal{S}} \mu_t(Z), \\ V_s &= \mathbb{E}\sum_{t\in\mathcal{S}_s(Z)} G_t. \end{aligned} \tag{8}$$

Then

$$V_{\mathrm{or}} - V_s = (V_{\mathrm{or}} - V_Z) + (V_Z - V_s), \qquad V_{\mathrm{or}} \ge V_Z \ge V_s. \tag{9}$$

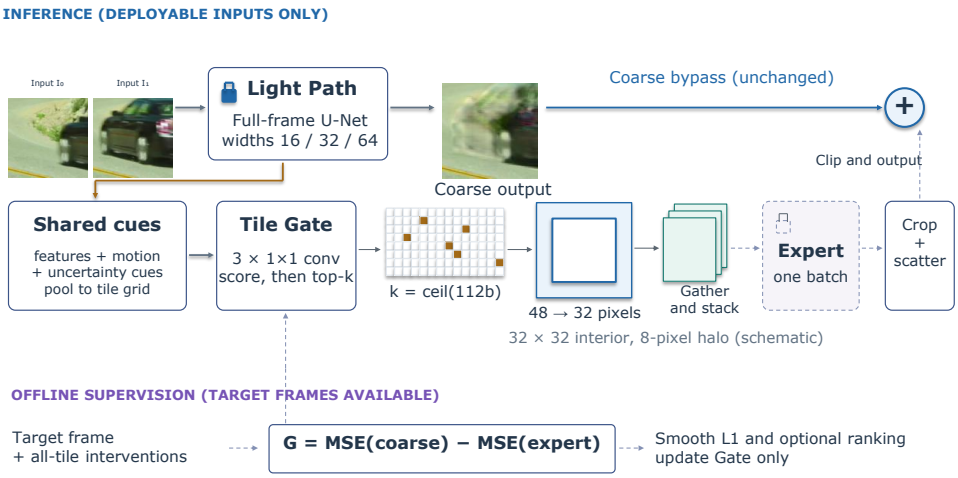


Figure 2: Fixed intervention, two information regimes. Online, deployable features score tiles and the selected halo crops pass through one frozen Expert batch before their interiors are composed with the coarse frame. Offline, target frames measure signed gain for the same interventions and supervise only the Gate. The Light Path, Expert, crop geometry, and exact tile budget are unchanged across routers.

The first term reflects information unavailable in Z; the second includes approximation, estimation, and optimization error of the Gate. Exhaustive gains establish full-information intervention headroom, while learned comparisons test how much of that value the selected features and training procedure capture. This decomposition concerns expected gain, equivalently expected MSE reduction, rather than expected PSNR. If scores estimate μ_t with uniform error at most ϵ, selected-gain regret relative to the best Z-based decision is at most $2k\epsilon$. Appendix A gives the proof and corresponding frame-MSE bound.

3.5 Gain-supervised Gate variants

Gain-supervised Gates use the same inputs and architecture as the Teacher-Advantage Gate. Only Gate weights are optimized. Smooth-L_1 regression uses scaled signed targets; the ranking variant adds a pairwise term:

$$\mathcal{L}_{\text{reg}} = N^{-1} \sum_t \text{smoothL1}(s_t, \alpha G_t), \tag{10}$$

$$\mathcal{L} = \mathcal{L}_{\text{reg}} + \lambda |\mathcal{P}|^{-1} \sum_{(i,j)\in\mathcal{P}} \text{softplus}\left[-\text{sign}(G_i - G_j)(s_i - s_j)\right]. \tag{11}$$

Pairs belong to the same image and satisfy $|G_i - G_j| > \delta$. A batch with no valid pairs contributes a zero ranking term. Label generation evaluates the frozen Expert with the same tile geometry as inference and detaches its outputs; the Gate forward pass remains differentiable. Regression-only and regression-plus-ranking are evaluated as separate variants, with training settings in Appendix B. These losses are surrogates for exact-budget utility: smooth-L_1 need not estimate the conditional mean in Equation 8, and pairwise loss includes comparisons outside the selected prefix. Appendix A.2 separates these objective properties from the empirical comparison.

4 Experimental Setup

Data and evaluation roles. We use Vimeo90K triplets (Xue et al., 2019). Vimeo500 is development data because it informed architecture and checkpoint selection. A separate manifest of 3,282 additional identifiers was frozen before evaluation; 3,281 have all required local frame files. The

locked comparison evaluates Motion, Uncertainty, the Teacher-Advantage Gate, and three Gain-Gate seeds on the same frozen Light–Expert configuration at exact budgets $k = 6$ and $k = 12$. No checkpoint, feature, or budget selection uses these locked outcomes. Split construction, the single unavailable identifier, and leakage controls are detailed in Appendix B.

Models and interventions. The sparse system contains a Light Path, shared-feature Gate, and local Expert, totaling 0.288M parameters. Router comparisons freeze predictions, Expert, and halo geometry. Motion and Uncertainty are proxy comparators; the deployable TA Gate learns Teacher-Advantage labels, whereas the diagnostic Teacher oracle ranks with the target frame. Gain regression has one screening seed; regression plus ranking uses seeds 2028, 2029, and 2030. Both oracles execute the same local Expert. On 256×448 images, 32×32 interiors give $N = 112$. Requested budgets of 2, 5, 10, and 20% select $k = 3, 6, 12, 23$, corresponding to 2.68, 5.36, 10.71, and 20.54% coverage. Six and twelve tiles are the core learned-routing budgets.

Training controls. The staged predictor is trained for three epochs with Adam on at most 10,000 examples. Gain variants share the same frozen Light and Expert weights; only the Gate changes, and the three selected regression-plus-ranking seeds use AdamW. Crop, batch, learning-rate, and target-scaling details appear in Appendix B.

Locked-set training-path audit. Source- and checkpoint-level audits verify that only Gate parameters change during gain-supervised training; Appendix B gives the implementation checks and their scope.

Exploratory mechanism checks. We transfer the preselected routers to all 310 triplets in each SNU-FILM tier (Choi et al., 2020), bilinearly resized to 256×448. Separate ridge probes are fitted only on Vimeo500. We also retrain two smaller Experts while keeping the Light Path fixed to test gain-map dependence on the action. These resized transfer and single-seed perturbation protocols are detailed in Appendix D.

Metrics and uncertainty. We report RGB PSNR on $[0, 1]$, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). PSNR gain is the mean per-image change from the common coarse output. Negative Selection Rate is the proportion of selected tiles with $G_t < 0$. Across-seed dispersion is standard deviation. For the locked comparison, we average the three Gain-Gate per-image PSNR gains within each image, subtract the paired Uncertainty gain, and bootstrap the 3,281 image-level differences (10,000 replicates); this interval quantifies image-sampling uncertainty for the three-seed average, not seed uncertainty. Appendix B gives metric definitions and the development-set interval procedure.

CPU measurement. CPU latency is measured on an Intel Core i5-13500H with eight PyTorch threads, using six fresh processes and 1,200 timed calls per model. The measured boundary includes routing, crop handling, Expert execution, composition, and clipping. The dense reference is selected within a predeclared 10% median-latency band; full software, warm-up, pooling, and exclusion details appear in Appendix B, and local-stage ablations are reported separately.

5 Results

5.1 Where does the fixed Expert offer useful intervention value?

The exhaustive development analysis contains 56,000 tile interventions. The frozen Expert reduces error on 62.41% and increases it on 37.59%. A selected tile can therefore have two costs: it consumes a call and, when its gain is negative, degrades the current prediction. Positive total improvement and a substantial harmful-call rate can coexist, which motivates evaluating selections rather than only the final mean image score.

The four-budget sweep in Appendix C (Figure 4) compares the same local actions. Motion, Uncertainty, and the Teacher-Advantage (TA) Gate form a close group, whereas the target-aware Actual-gain oracle selects interventions with larger realized gain. At six tiles, the Teacher oracle gives

0.1576 dB and the gain oracle 0.2428 dB. Both use the target frame; their 0.0852 dB difference shows that teacher improvement is not the same ranking target as gain from the deployed Expert.

The core-budget evaluation provides the corresponding action-level view. At six tiles, Uncertainty selects harmful interventions in 20.733% of its calls, versus zero for the gain oracle. At twelve tiles, the oracle gains 0.33380 dB compared with 0.20133 dB for Uncertainty, and its harmful-call rate is 0.167%. An exact budget is important: images with fewer than twelve positive-gain tiles force the oracle into the non-positive tail, even though the aggregate population contains many beneficial tiles. Thus, the global positive fraction alone does not determine the safest per-image budget.

5.2 Does direct gain supervision capture the oracle margin?

Direct supervision changes the training target while preserving the Gate's inputs and architecture. Table 1 compares the resulting routers at the two core budgets.

Table 1: Exact-budget learned routing on the 500-example Vimeo development split. Gain regression + ranking is the three-seed mean ($\pm$ standard deviation); regression-only has one screening seed. The oracle uses target frames.

	$k = 6$ (5.36%)		$k = 12$ (10.71%)	
Router	Gain (dB)	Harmful (%)	Gain (dB)	Harmful (%)
Motion	.12926	21.600	.20045	21.600
Uncertainty	.13225	20.733	.20133	21.217
TA Gate	.13145	21.267	.20168	21.417
Gain regression	.13193	21.167	.20161	21.750
Gain regression + ranking	.13018$\pm$.00005	21.233	.20214$\pm$.00003	21.594
Gain oracle	.24276	.000	.33380	.167

The tested losses stay close to Uncertainty. Regression alone changes PSNR by -0.00032 and $+0.00028$ dB at six and twelve tiles. Regression plus ranking changes it by -0.00207 and $+0.00081$ dB, with across-seed standard deviations of 0.00005 and 0.00003 dB. The ranking-augmented variant is higher at twelve tiles and lower at six tiles than the regression-only screening run.

The seed dispersion is small, but paired image-bootstrap intervals for every seed include zero at both budgets. Stability across training seeds therefore coexists with uncertainty about the mean advantage across images. Changing the label and adding generic pairwise ordering yield small PSNR differences without reducing the harmful-call rate. The aggregate comparison concerns reconstruction quality and harmful-call frequency. Appendix C reports the supplementary records.

5.3 Does the routing result persist on a frozen locked set?

We compare all routers without tuning on the same 3,281 locked examples, frozen Light–Expert pair, and exact budgets. Figure 3 reports each Gain-Gate seed and the three-seed image-paired bootstrap interval against Uncertainty.

The paired mean difference is -0.000520 dB at $k = 6$ (95% CI $[-0.001694, 0.000663]$) and $+0.000505$ dB at $k = 12$ (95% CI $[-0.000640, 0.001653]$). Both intervals span zero, and their upper endpoints are below 0.002 dB for the tested three-seed average. A post-hoc native-objective analysis is unresolved at six tiles but finds a positive mean-MSE contrast at twelve; Appendix B reports the transformation and paired intervals. Thus, mean-MSE and mean-PSNR aggregation can support different resolutions. The corresponding harmful-call rates are 18.93% and 20.06%, versus 18.74% and 19.46% for Uncertainty. At twelve tiles, a slightly higher PSNR estimate coexists with more harmful calls; frequency alone does not measure their magnitude. These intervals quantify the tested paired means, not equivalence or the unrestricted feature-conditioned optimum V_Z.

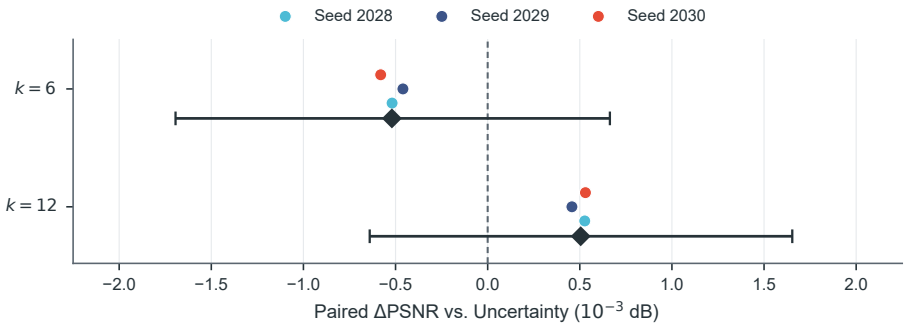


Figure 3: Gain-Gate versus Uncertainty on the frozen 3,281-example locked set at exact budgets $k \in \{6, 12\}$. Colored points are per-seed paired mean PSNR differences; black diamonds and horizontal intervals show the three-seed average and image-paired bootstrap 95% CI. The dashed line denotes no difference; both intervals include zero.

5.4 Exploratory mechanism checks: transfer and Expert dependence

Transfer at two exact budgets. Table 2 compares frozen Gain Gates with Uncertainty on each resized SNU-FILM tier. At six tiles, all four image-bootstrap intervals span zero. At twelve tiles, the Easy, Medium, and Hard intervals lie below zero, while Extreme remains unresolved. These exploratory, unadjusted comparisons show lower relative PSNR in three tiers at $k = 12$; the separate intervals do not test the difference between budgets. A direct image-paired contrast resolves that difference only for Medium, not for Easy, Hard, or Extreme (Appendix D). The original Expert retains target-aware headroom over Uncertainty: 0.0784–0.0824 dB at six tiles and 0.0878–0.0994 dB at twelve tiles.

Table 2: Frozen Gate transfer to resized SNU-FILM. Entries are paired PSNR differences from Uncertainty after averaging the three Gain-Gate seeds within each image, in 10^{-3} dB, with image-bootstrap 95% intervals. Each tier has 310 images at 256×448; intervals are exploratory and unadjusted for multiple comparisons.

Tier	$k = 6$: mean [95% CI]	$k = 12$: mean [95% CI]
Easy	-2.2 $[-6.5,\ 2.1]$	-5.6 $[-9.9, -1.3]$
Medium	0.9 $[-2.4,\ 4.1]$	-5.9 $[-9.2, -2.7]$
Hard	-0.3 $[-3.7,\ 2.9]$	-3.5 $[-6.7, -0.5]$
Extreme	1.1 $[-1.5,\ 3.6]$	0.1 $[-2.3,\ 2.6]$

Feature enrichment and Expert dependence. The Vimeo500-fitted endpoint/RGB probe changes transfer PSNR by -0.0007 to $+0.0063$ dB across the eight tier–budget settings. Shared-feature probes incur larger losses on Easy and Medium, reaching -0.0546 dB on Easy at twelve tiles. Thus, feature enrichment does not uniformly improve these fitted linear probes. Changing the Expert also changes the target: on Vimeo500, the width-16 deep and width-32 shallow perturbations have gain-map correlations of 0.486 and 0.403 with the original, and harmful-tile fractions of 58.1% and 56.1%, compared with 37.6%. Their positive oracle margins of 0.0989 and 0.0686 dB coexist with lower absolute refinement quality. Appendix D gives the full probe and perturbation results.

5.5 What changes when execution and context are separated?

The looped and vectorized halo variants have zero maximum output difference in the reported comparison. Their median local-stage times are 44.31 and 27.37 ms, and their p95 times are 83.39 and 35.37 ms. The $1.62\times$ median speedup therefore comes from executing the same selected work differently. Removing the halo reduces the vectorized local-stage median to 15.61 ms but changes reconstruction quality. Relative to the common coarse frame, halo refinement adds 0.1315 dB, compared with 0.0404 dB without the halo; the 0.0911 dB difference quantifies the role of additional

input context for this fixed configuration (Table 7). Together, the two ablations separate a scheduling gain that preserves the intervention from a context reduction that changes its reconstruction value.

5.6 How does the sparse system compare at measured CPU cost?

End-to-end sparse and dense medians fall within the predeclared 10% matching band; Table 3 reports the pooled quality and latency measurements.

Table 3: Vimeo500 development quality and fixed-host CPU latency. Latency pools six independent processes across two rounds (1,200 timed calls per model); lower LPIPS is better.

Model	Params (M)	Median (ms)	p95 (ms)	PSNR (dB)	SSIM	LPIPS
Dense matched	.487	229.70	331.23	28.5748	.8509	.08577
Sparse	.288	241.00	364.44	29.2754	.8651	.08142

The latency ordering changes between rounds. At the pooled operating points, sparse development quality is higher by approximately 0.701 dB PSNR and 0.01424 SSIM, with LPIPS lower by 0.00435. Paired bootstrap 95% intervals are $[0.6548, 0.7470]$ dB, $[0.0130, 0.0155]$, and $[0.00356, 0.00516]$, respectively.

Figure 5 in Appendix C decomposes the printed development PSNR means: 0.5691 dB of the dense-to-sparse gap predates refinement, while the fixed-Light refinement adds 0.1315 dB. The dense reference and Light Path differ in training and architecture, so the comparison supports a higher-quality sparse development point at near-matched latency, not a causal attribution of the full gap to routing or a sparse-speed advantage.

6 Discussion and Limitations

Intervention value and predictable routing value. The development oracle measures full-information value, whereas the locked comparison evaluates input-only policies. Source and checkpoint audits confirm Gate updates with a fixed Light–Expert pair. The near-tie with Uncertainty can still reflect feature information, surrogate calibration, or model fitting; the fitted probes do not estimate the unrestricted feature-conditioned optimum V_Z. Thus, the evidence localizes the gap to the deployed feature–surrogate–model combination without isolating a single cause.

Negative gain makes allocation asymmetric: a false positive costs computation and quality, whereas a false negative forgoes a possible gain. Negative Selection Rate complements net PSNR because it counts harmful calls without measuring their magnitude. Even the exact-budget oracle enters the non-positive tail when fewer than k beneficial calls exist, so population statistics do not define a safe per-image budget. Adaptive or at-most-k routing is therefore a natural extension, but it requires a stopping decision in addition to ranking.

Scope. The locked comparison uses one Light–Expert pair. Transfer uses resized SNU frames and perturbations use one training seed per Expert; a Gate is not retrained for each perturbation. CPU latency remains host- and protocol-specific.

7 Conclusion

For the studied Light–Expert configuration, signed Actual Expert Gain exposes useful and harmful local refinements and substantial full-information allocation headroom. Across development and locked data, the tested gain-supervised Gates remain close to endpoint disagreement in mean PSNR; resized-SNU results show that this comparison is budget- and domain-sensitive, while Expert perturbations change the gain target itself. Output-matched vectorization accelerates the local stage without implying an end-to-end sparse speed advantage. Overall, sparse refinement separates intervention value, predictability from deployable features, and execution cost.

AI USE STATEMENT

Generative AI tools were used to polish the language and improve grammatical clarity and readability. All scientific content, methods, analyses, results, and conclusions were developed and verified by the authors.

REFERENCES

W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang. Depth-aware video frame interpolation. In *CVPR*, pages 3703–3712, 2019.

R. Cheng, X. Jiang, R. He, S. Zhou, W. Tan, and B. Yan. Uncertainty-guided spatial pruning architecture for efficient frame interpolation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1975–1986, 2023. doi:10.1145/3581783.3611752.

M. Choi, H. Kim, B. Han, N. Xu, and K. M. Lee. Channel attention is all you need for video frame interpolation. In *AAAI*, pages 10663–10671, 2020.

M. Choi, S. Lee, H. Kim, and K. M. Lee. Motion-aware dynamic architecture for efficient frame interpolation. In *ICCV*, pages 13839–13848, 2021.

J. Dekoninck, M. Baader, and M. Vechev. A unified approach to routing and cascading for LLMs. In *ICML*, PMLR 267:12987–13010, 2025.

A. N. Elmachtoub and P. Grigas. Smart "Predict, then Optimize". *Management Science*, 68(1):9–26, 2022.

A. Heakl, M. Gubri, S. Khan, S. Yun, and S. J. Oh. Dr.LLM: Dynamic layer routing in LLMs. In *ICLR*, 2026.

P. Hu, S. Niklaus, L. Zhang, S. Sclaroff, and K. Saenko. Video frame interpolation with many-to-many splatting and spatial selective refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2):823–836, 2024.

P. Hu, S. Niklaus, S. Sclaroff, and K. Saenko. Many-to-many splatting for efficient video frame interpolation. In *CVPR*, pages 3553–3562, 2022.

Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou. Real-time intermediate flow estimation for video frame interpolation. In *ECCV*, pages 624–642, 2022.

H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, 2018.

W. Jitkrittum, N. Gupta, A. K. Menon, H. Narasimhan, A. S. Rawat, and S. Kumar. When does confidence-based cascade deferral suffice? In *NeurIPS*, volume 36, 2023.

L. Kong, B. Jiang, D. Luo, W. Chu, X. Huang, Y. Tai, C. Wang, and J. Yang. IFRNet: Intermediate feature refine network for efficient frame interpolation. In *CVPR*, pages 1969–1978, 2022.

Z. Li, Z.-L. Zhu, L.-H. Han, Q. Hou, C.-L. Guo, and M.-M. Cheng. AMT: All-pairs multi-field transforms for efficient frame interpolation. In *CVPR*, pages 9801–9810, 2023.

C. Liu, G. Zhang, R. Zhao, and L. Wang. Sparse global matching for video frame interpolation with large motion. In *CVPR*, pages 19125–19134, 2024.

A. Lv, J. Ma, Y. Ma, and S. Qiao. Coupling experts and routers in mixture-of-experts via an auxiliary loss. In *ICLR*, 2026.

N. Ma, X. Zhang, H.-T. Zheng, and J. Sun. ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In *ECCV*, pages 116–131, 2018.

Y. Ma, S. Zhao, M. Yao, J. Li, L. Zhang, X. Liu, Q. Dou, J. Gu, and T. Xue. Realtime video frame interpolation using one-step diffusion sampling. In *ICLR*, 2026.

H. Mozannar and D. Sontag. Consistent estimators for learning to defer to an expert. In *ICML*, pages 7076–7087, 2020.

H. Narasimhan, W. Jitkrittum, A. S. Rawat, S. Kim, N. Gupta, A. K. Menon, and S. Kumar. Faster cascades via speculative decoding. In *ICLR*, 2025.

S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, pages 261–270, 2017.

J. Puigcerver, C. Riquelme Ruiz, B. Mustafa, and N. Houlsby. From sparse to soft mixtures of experts. In *ICLR*, 2024.

Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, pages 13937–13949, 2021.

M. S. Ryoo, A. J. Piergiovanni, A. Arnab, M. Dehghani, and A. Angelova. TokenLearner: Adaptive space-time tokenization for videos. In *NeurIPS*, 2021.

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Y. Wang, Y. Yue, Y. Lin, H. Jiang, Z. Lai, V. Kulikov, N. Orlov, H. Shi, and G. Huang. AdaFocus V2: End-to-end training of spatial dynamic networks for video recognition. In *CVPR*, pages 20062–20072, 2022.

W. Wang, Q. Wang, K. Zheng, H. Ouyang, Z. Chen, B. Gong, H. Chen, Y. Shen, and C. Shen. Framer: Interactive frame interpolation. In *ICLR*, 2025.

T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019.

L. Yuan, Y. Chen, H. Liu, T. Kong, and J. Shi. Zoom-In-To-Check: Boosting video interpolation via instance-level discrimination. In *CVPR*, pages 12183–12191, 2019.

R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.

Z. Zhong, G. Krishnan, X. Sun, Y. Qiao, S. Ma, and J. Wang. Clearer Frames, Anytime: Resolving velocity ambiguity in video frame interpolation. In *ECCV*, pages 346–363, 2024.

A Fixed-Budget Analysis

The baseline called Uncertainty is algebraically related to endpoint motion. For each channel,

$$|c-a| + |c-b| = |a-b| + 2\,\mathrm{dist}(c, [\min(a,b), \max(a,b)])\,, \tag{12}$$

and therefore

$$u(x) = \tfrac{1}{2} m(x) + \tfrac{1}{3} \sum_c \mathrm{dist}(I_c^c(x), [\min(I_0^c, I_1^c), \max(I_0^c, I_1^c)])\,. \tag{13}$$

When the coarse channel lies between the two endpoints, the second term vanishes. This overlap can make Motion and Uncertainty behave similarly, although we do not assume that the condition holds frequently.

Let the frame contain N equal-area, non-overlapping tile interiors. Write $e_{c,t}$ for the tile-mean MSE of the coarse output and $e_{e,t}$ for the MSE after applying the frozen Expert to tile t. By definition, $\mathcal{G}_t = e_{c,t} - e_{e,t}$. If the Expert output for one tile does not depend on which other tiles are evaluated in the same batch, then selecting a set $\mathcal{S}$ changes only those tile interiors:

$$E(\mathcal{S}) = \frac{1}{N}\left(\sum_{t \notin \mathcal{S}} e_{c,t} + \sum_{t \in \mathcal{S}} e_{e,t}\right) \tag{14}$$

$$= E_c - \frac{1}{N} \sum_{t \in \mathcal{S}} \mathcal{G}_t. \tag{15}$$

For an exact budget $|\mathcal{S}| = k$, minimizing MSE is therefore equivalent to selecting the k largest signed gains. This is the objective optimized by the Actual-gain oracle. The result does not extend automatically to overlapping output interiors, Experts whose outputs interact across selected tiles, or non-additive metrics such as SSIM and LPIPS.

The same independence condition permits all tile labels to be generated in batches: batching changes execution order but not the intervention assigned to a tile. An exact-k policy can select a negative-gain tile when fewer than k positive gains exist. By contrast, a policy with an upper-bound budget may stop after the positive prefix. Clipping negative gains to zero preserves the order among positive tiles, but removes the distinction between neutral and harmful calls and cannot represent the ordering of the non-positive tail.

A.1 Information-conditioned value decomposition

Let Z contain all information visible to the Gate, including the spatial arrangement of its features, and let $\mu_t(Z) = \mathbb{E}[G_t \mid Z]$. For any Z-measurable selection rule $\mathcal{S}(Z)$,

$$\mathbb{E}\left[\sum_{t\in\mathcal{S}(Z)} G_t\right] = \mathbb{E}\left[\mathbb{E}\left[\sum_{t\in\mathcal{S}(Z)} G_t \mid Z\right]\right] \tag{16}$$

$$= \mathbb{E}\left[\sum_{t\in\mathcal{S}(Z)} \mu_t(Z)\right]. \tag{17}$$

Thus, selecting the k largest conditional means maximizes expected selected gain among Z-measurable rules, and $V_Z \geq V_s$. Conditional expectation and maximization also give

$$\mathbb{E}\left[\max_{|\mathcal{S}|=k} \sum_{t\in\mathcal{S}} G_t \mid Z\right] \geq \max_{|\mathcal{S}|=k} \sum_{t\in\mathcal{S}} \mathbb{E}[G_t \mid Z], \tag{18}$$

so $V_{\text{or}} \geq V_Z$. Adding and subtracting V_Z yields Equation 9. Both terms are non-negative, but the observed oracle-to-policy difference alone does not determine either term. In particular, a failed training procedure is not an estimate of V_Z.

Score-error bound. For a fixed feature realization, let $\mathcal{S}_Z^*$ select the k largest $\mu_t(Z)$ and let $\mathcal{S}_s$ select the k largest scores s_t. If $\max_t |s_t - \mu_t(Z)| \leq \epsilon$, then

$$0 \leq \sum_{t\in\mathcal{S}_Z^*} \mu_t(Z) - \sum_{t\in\mathcal{S}_s} \mu_t(Z) \leq 2k\epsilon. \tag{19}$$

The bound follows by adding and subtracting the selected score sums. Optimality of $\mathcal{S}_s$ makes the score difference non-positive, leaving at most $k\epsilon$ from each selection's estimation error. Dividing by N gives a conditional expected frame-MSE regret of at most $2k\epsilon/N$. Scores in this bound are expressed in gain units. The analysis is not fitted to the reported data or presented as an additional empirical measurement.

A.2 Surrogate objectives and conditional gain

The decision rule in Section 3.4 uses the conditional mean $\mu_t(Z)$. This is an objective-level characterization, not a consistency claim for the fitted Gate loss. With the default unit transition of smooth-L_1, write $Y = \alpha G_t$ and let $a(Z)$ minimize conditional regression risk. Its first-order condition is

$$\mathbb{E}[\text{clip}(a(Z) - Y, -1, 1) \mid Z] = 0. \tag{20}$$

The clipped residual equation generally differs from $a(Z) = \mathbb{E}[Y \mid Z]$; they agree when the relevant residuals stay in the quadratic region. Adding pairwise loss further changes the objective. In addition, ranking two tiles already inside the selected set, or two already outside it, leaves the exact-k allocation unchanged, although both comparisons can contribute to the training loss. Thus, action-aligned labels, conditional-mean estimation, and budget-focused ranking are separate design choices. These properties motivate controlled comparisons of losses but do not identify the empirical cause of the reported near-tie.

B Full Experimental Protocol

B.1 Data roles and leakage control

The historical Vimeo500 subset has already influenced architecture and checkpoint selection and is therefore development data. From the 3,782 valid examples in the official test index, the remaining 3,282 non-overlapping identifiers form a locked subset. Its manifest was frozen before the final evaluation (SHA-256: `6f42d9e8a5ba856d86cb098aa795c85c29ab7f5ab7f76a966a9e62ac2e10a72c`). Identifier `00060/0479` is missing all required frame files in the local dataset copy. We retain the frozen manifest and evaluate the remaining 3,281 examples. The same Light and Expert are used to compare Motion, Uncertainty, the Teacher-Advantage Gate, and Gain-Gate checkpoints from seeds 2028, 2029, and 2030 at exact $k = 6$ and $k = 12$; per-router results appear in Table 4. No parameters, feature sets, or budgets are selected using these results. This is a router comparison for one frozen architecture, not a cross-Expert or dense-model comparison.

External results enter the final comparison only when image range, metric code, input resolution, checkpoint, and timing boundary match the frozen protocol. Earlier heterogeneous runs are not combined with the development tables in the main paper.

B.2 Training settings

The staged checkpoint uses Light and Expert base widths of 16 and 32. It was trained on at most 10,000 training samples with 128×128 crops, batch size 16, Adam optimization, an initial learning rate of 10^{-4}, three epochs, and a fixed seed of 2028. Tile size, selection ratio, and halo are 32, 5%, and eight pixels. The Light Path is frozen in the final staged run.

Gain-supervised experiments start from the same frozen Light and Expert weights; only the 593-parameter Gate is optimized. The implementation produces a fully refined image during training, pools the signed coarse-to-refined MSE difference into tile targets, and applies smooth-L_1 regression. The final regression-plus-ranking runs use one epoch, AdamW, batch size eight, learning rate 2×10^{-4}, weight decay 10^{-4}, target scale 1,000, ranking weight 0.1, and target-pair margin 10^{-7}, without a learning-rate scheduler. The regression-only screening run uses one seed. Seeds 2028, 2029, and 2030 share the same frozen Light–Expert starting checkpoint and select the single-epoch checkpoint by the mean development PSNR gain at the two core budgets. They are then evaluated without tuning on the locked manifest. The locked router run uses PyTorch 2.12.0+cpu, eight CPU threads, exact top-k, and one shared Light/Expert forward per sample; selected-tile unions are reused to avoid repeated Expert computation across routers. Per-sample predictions and checkpoint hashes are included with the experiment records.

Gradient-path audit. In `train_expert_gain_gate.py`, the frozen-target routine is under `@torch.no_grad()`, while Gate scoring occurs afterward in the ordinary differentiable forward path. All model parameters are first frozen, then only `model.gate` parameters are re-enabled. The optimizer is constructed from Gate parameters only, and the regression/ranking loss is followed by `backward()` and an optimizer step. Checkpoint tensor comparisons for the three locked Gain-Gate seeds show Gate changes and no Light/Expert changes. The training logs do not contain per-step gradient norms; consequently, this audit verifies source-level gradient boundaries and checkpoint-level updates but cannot establish gradient magnitudes at every step.

B.3 Metric definitions

Images are evaluated in RGB on $[0, 1]$. PSNR gain is averaged over images relative to the corresponding coarse output. Negative Selection Rate is the fraction of selected tiles with $\mathcal{G}_t < 0$. For Expert perturbations, Spearman correlation compares the original and perturbed signed-gain rankings within each image, followed by averaging over images. SSIM and LPIPS are reported for the dense–sparse system comparison. Historical Gain Capture, Oracle Efficiency, and NDCG fields are not used as evidence for the primary conclusions; their provenance is discussed under Precision and aggregation.

For dense–sparse comparison, we bootstrap per-example paired differences and report percentile 95% intervals. LPIPS is oriented as dense minus sparse so that a positive reported advantage denotes a lower, better sparse value.

For development Gain Gate versus Uncertainty, each seed is compared on the same 500 development examples; these per-seed percentile bootstrap intervals resample paired examples and are not intervals over seeds. For the locked comparison, define $d_i = \frac{1}{3}\sum_{s=1}^{3} \Delta\mathrm{PSNR}_{i,s} - \Delta\mathrm{PSNR}_{i,U}$ for each of the 3,281 images. We resample these paired image-level differences with replacement for 10,000 bootstrap replicates (analysis seed 20260918). Averaging concerns scalar PSNR gains, not predicted images, and therefore does not construct an ensemble interpolator. The resulting intervals quantify paired image-sampling uncertainty for the three-seed average and do not treat repeated seeds as independent images. The analysis is reproducible with `03_scripts/summarize_locked_gain_gate.py`; its output is `gain_gate_vs_uncertainty.csv`.

Table 4: Locked router comparison. Gain denotes PSNR improvement over the common coarse output; NSR is the harmful-call fraction. Each router is evaluated on the same 3,281 examples at the exact tile budget.

	$k = 6$		$k = 12$	
Router	Gain (dB)	NSR (%)	Gain (dB)	NSR (%)
Motion	.146697	18.932	.224981	19.994
Uncertainty	.148494	18.739	.226970	19.455
Teacher-Advantage Gate	.148209	18.912	.226948	19.803
Gain Gate seed 2028	.147975	18.937	.227497	20.060
Gain Gate seed 2029	.148034	18.897	.227427	20.052
Gain Gate seed 2030	.147913	18.947	.227501	20.055

Table 5: Exploratory native-objective reanalysis on the locked set. Entries are Gain Gate minus Uncertainty after averaging the three fixed seeds within each image. Positive MSE values mean that the Gain Gate achieves a larger reduction in RGB $[0, 1]$ frame MSE. Each seed's MSE reduction is recovered from its per-image PSNR gain and the corresponding coarse MSE before averaging; intervals resample paired images 10,000 times.

Contrast	PSNR (dB), mean [95% CI]	MSE reduction (10^{-7}), mean [95% CI]
$k = 6$	$-.000520$ $[-.001694, .000663]$	5.45 $[-1.04, 12.11]$
$k = 12$	$.000505$ $[-.000640, .001653]$	8.18 $[1.75, 14.90]$
$k = 12$ minus $k = 6$	$.001025$ $[-.000436, .002516]$	2.73 $[-5.44, 11.03]$

The first two rows compare routers at a fixed budget; the last row pairs each image across budgets and tests the change in relative performance. This analysis was added after inspection and is unadjusted. It therefore clarifies the relation between the additive MSE objective and the reported image-mean PSNR without redefining the frozen locked comparison as confirmatory.

B.4 CPU TIMING AND DENSE SEARCH

Measurements run in evaluation and inference modes with eight PyTorch intra-op threads, batch size one, and 256×448 inputs. Each fresh process performs 50 warm-up and 200 timed iterations. Each architecture was measured in three independent processes per round across two rounds (six processes and 1,200 timed calls in total). The sparse timing boundary begins before the Light Path and ends after composition; it includes Gate inference, topk, halo gather, the batched Expert, scatter, and clipping. Image decoding, file I/O, and metric computation are excluded.

The dense model is expanded and remeasured until its median lies within 10% of the sparse median. This latency match is preferred to parameter or FLOP matching because sparse indexing and dispatch are not represented by either surrogate. Pooling all six processes gives the 0.487M dense candidate a 229.70 ms median and 331.23 ms p95; the 0.288M sparse system measures 241.00 ms and 364.44 ms. The per-round latency ordering changes, so these results support a near-latency-matched comparison rather than a sparse speed advantage. The machine-readable records report round-specific values and all per-call timings.

B.5 Precision and aggregation

Reported means retain their source precision. The four-budget sweep and the higher-precision core-budget evaluation are distinct records, including last-digit differences in proxy scores. Differences computed within a table use that table's means; the dense–Light–Sparse decomposition uses its printed development PSNRs. The reported paired dense–sparse estimates and intervals are retained at their supplied precision. Primary routing comparisons use PSNR and the directly defined harmful-call rate. Archived Gain Capture, Oracle Efficiency, and NDCG summaries are secondary records rather than inputs to the main conclusions because the earlier aggregation and relevance-transform details are not fully preserved.

C Additional Experimental Records

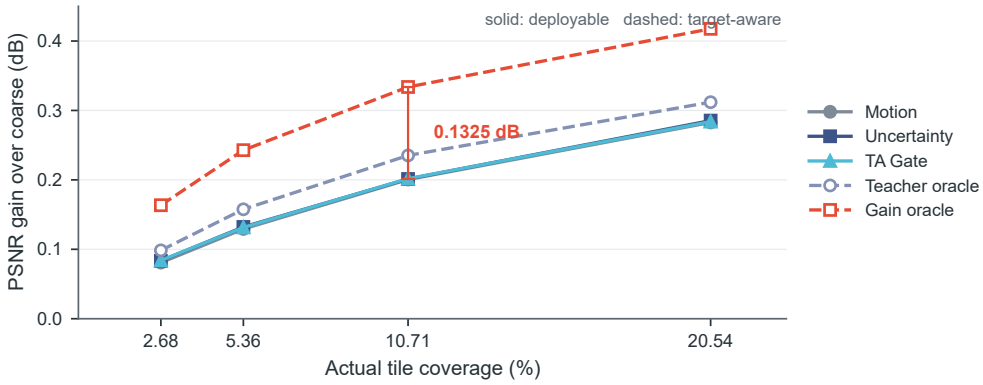


Figure 4: Routing gains across exact tile budgets on Vimeo500. Solid curves are input-only routers; dashed curves are target-aware oracles. Coverage reflects integer tile counts. This sweep is a separate record from the higher-precision core-budget evaluation in Table 1.

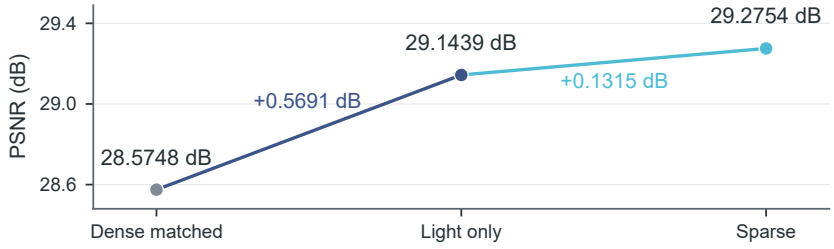


Figure 5: The dense–sparse gap is not a pure routing effect. Values are derived from printed development means; only the Light-to-Sparse step holds the coarse model fixed.

Table 6: PSNR improvement (dB) over the common coarse prediction on Vimeo500. Requested budgets map to 3, 6, 12, and 23 tiles, or 2.68, 5.36, 10.71, and 20.54% actual coverage.

Router	2.68%	5.36%	10.71%	20.54%
Motion	.0808	.1293	.2005	.2827
Uncertainty	.0834	.1322	.2013	.2854
TA Gate	.0840	.1315	.2017	.2839
Teacher oracle	.0985	.1576	.2352	.3119
Actual-gain oracle	**.1635**	**.2428**	**.3338**	**.4176**

For the three-seed ranking Gate, the paired image-bootstrap 95% intervals include zero for every seed at both budgets. The standard deviations across seeds are 0.00005 dB at $k = 6$ and 0.00003 dB at $k = 12$. These quantities have different sampling units and are reported separately.

Table 7: Local-stage context and execution ablation.

Variant	PSNR (dB)	Gain (dB)	Median (ms)	p95 (ms)
Loop, halo 8	29.2754	.1315	44.31	83.39
Vectorized, halo 8	29.2754	.1315	27.37	35.37
Vectorized, no halo	29.1843	.0404	15.61	21.04

D Exploratory Mechanism Checks

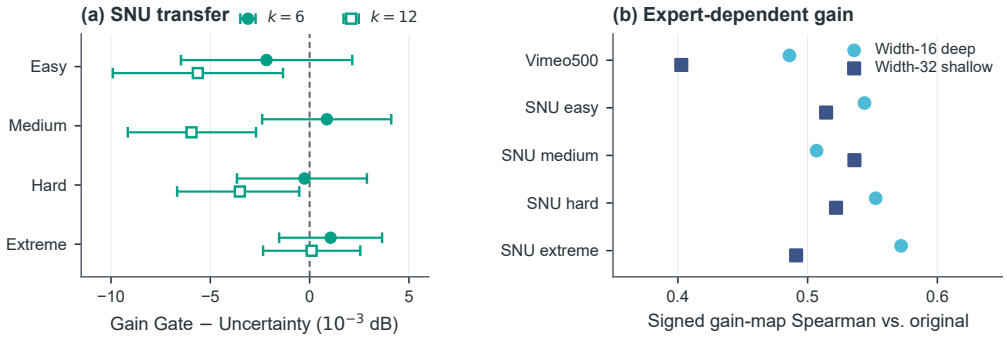


Figure 6: Exploratory transfer and Expert dependence. Left: three-seed Gain-Gate mean minus Uncertainty on resized SNU-FILM at $k = 6$ (filled circles) and $k = 12$ (open squares), with separate paired image-bootstrap 95% intervals. Right: per-image Spearman correlation between the original Expert's signed gain map and each trained perturbation, averaged by dataset. All SNU frames are resized to 256×448; neither plot is an official full-resolution benchmark.

Transfer protocol. We evaluate all 310 triplets in each official SNU-FILM easy, medium, hard, and extreme list after bilinear resizing to 256×448. No SNU target is used for fitting or selecting a router. The Teacher-Advantage Gate and Gain-Gate seeds 2028, 2029, and 2030 are the preselected frozen checkpoints from the main comparison. The reported ridge-probe analysis (ℓ_2 coefficient 1) covers two enriched feature sets fitted to signed tile gains on Vimeo500: Motion and Uncertainty plus 16 pooled Light channels, and Motion and Uncertainty plus pooled endpoint RGB, coarse RGB, and absolute endpoint difference. All methods select exactly six or twelve tiles. The oracle uses the target frame and is diagnostic. Table 8 reports paired PSNR differences from Uncertainty; confidence intervals resample images 10,000 times and are not adjusted for multiple comparisons. For the Gain Gate, scalar PSNR gains are averaged across seeds within each image before resampling; predicted RGB frames are not averaged. These results are resized mechanism comparisons, not official full-resolution SNU-FILM benchmark results.

Only the Medium-tier interval excludes zero. The separately reported $k = 12$ intervals for Easy and Hard therefore do not by themselves establish a budget interaction.

Expert perturbations. The original local Expert is a width-32 deep network with 95,875 Expert parameters. Keeping the Light Path fixed, we train a width-16 deep Expert (24,899 parameters) and a width-32 shallow Expert (12,739 parameters) for three epochs on 10,000 Vimeo90K training examples each, with seed 2028 and a 500-example validation set. The shallow version has one convolutional block and an output layer in place of the original encoder–decoder. The Gate stays frozen. Evaluation checks bit-identical Light weights and coarse outputs. Table 10 reports the mean per-image Spearman correlation between each new Expert's signed gain map and the original map,

Table 8: Resized SNU-FILM router transfer. Entries are mean paired PSNR differences from Uncertainty in dB. The Gain-Gate column averages scalar PSNR gains across three seeds within each image and includes image-bootstrap 95% confidence intervals; the other columns report point estimates. Full per-image records and intervals for every method are retained with the experiment artifacts.

Tier	k	Gain Gate, three seeds	Endpoint probe	Shared probe	Gain oracle
Easy	6	$-.0022$ $[-.0065, .0021]$	$-.0000$	$-.0393$	.0801
Easy	12	$-.0056$ $[-.0099, -.0013]$	.0004	$-.0546$	.0885
Medium	6	.0009 $[-.0024, .0041]$	.0031	$-.0136$	.0821
Medium	12	$-.0059$ $[-.0092, -.0027]$	$-.0007$	$-.0270$	.0878
Hard	6	$-.0003$ $[-.0037, .0029]$	.0063	$-.0018$	.0784
Hard	12	$-.0035$ $[-.0067, -.0005]$	.0032	$-.0115$	.0912
Extreme	6	.0011 $[-.0015, .0036]$	.0049	.0016	.0824
Extreme	12	.0001 $[-.0023, .0026]$	.0053	.0040	.0994

Table 9: Direct paired budget contrast on resized SNU-FILM. Each entry is the change in Gain-Gate-minus-Uncertainty PSNR from $k = 6$ to $k = 12$, in 10^{-3} dB, with an image-bootstrap 95% interval. Negative values mean that the Gate's relative performance decreases at the larger budget. Intervals are exploratory and unadjusted.

Tier	$(d_{12} - d_6)$, mean [95% CI]
Easy	-3.46 $[-8.95, \ 2.02]$
Medium	-6.82 $[-11.10, -2.64]$
Hard	-3.25 $[-6.75, \ 0.19]$
Extreme	-0.95 $[-3.99, \ 2.03]$

the fraction of tiles with negative gain, and each Expert's paired gain-oracle margin over Uncertainty at $k = 6$. The two perturbed Experts have lower absolute refinement quality than the original; the margins in the table are therefore not deployable improvements over the original system.

Table 10: Expert-dependent signed gain. All SNU rows use 256×448 resized frames. "Neg." is the fraction of harmful tile calls; "Oracle gap" is oracle minus Uncertainty PSNR gain in dB at $k = 6$. Correlations compare each perturbation with the original Expert. Each perturbed Expert has one training seed and is not latency matched.

	Original	Width-16 deep			Width-32 shallow		
Dataset	Neg. (%)	ρ	Neg. (%)	Oracle gap	ρ	Neg. (%)	Oracle gap
Vimeo500	37.6	.486	58.1	.0989	.403	56.1	.0686
SNU easy	58.7	.544	78.8	.0722	.514	80.3	.1280
SNU medium	46.4	.507	68.0	.0567	.536	66.6	.0579
SNU hard	41.4	.552	56.0	.0565	.522	54.6	.0438
SNU extreme	43.4	.572	52.4	.0734	.491	48.0	.0490

Both perturbations leave positive target-aware oracle headroom at each dataset and budget tested, but change the intervention rankings and harmful-call frequency. The experiment does not evaluate a retrained Gate for either Expert. It therefore supports Expert dependence of the measured target, not cross-architecture performance of gain-supervised routing. The source records include the exact checkpoint hashes, per-image scores, and paired confidence intervals.

E Qualitative Interventions

The examples show local blur or shape distortion reduced by the Expert while the rest of the coarse frame remains unchanged.

F Reference Operations

The routines below specify the gradient boundary, signed-label computation, and vectorized composition associated with the formulation. Frozen labels and trainable scores use separate forward

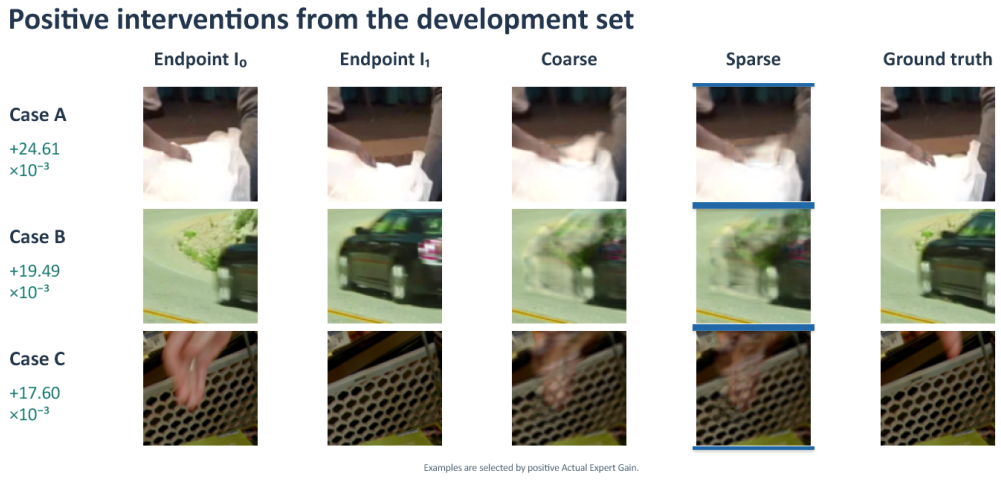


Figure 7: Positive local interventions. Rows are three selected cases; columns show endpoint I_0, endpoint I_1, coarse prediction, sparse refinement, and ground truth. Cases are selected by positive Actual Expert Gain for visualization; quantitative conclusions use all 500 development images.

paths: disabling gradients for label generation does not disable gradients through the Gate. Complete experiment scripts, input checks, and isolated unit tests accompany the reference operations.

F.1 Signed labels and Gate gradient boundary

Listing 1: Signed tile-MSE labels.

```python
@torch.no_grad()
def signed_gain(coarse, refined, target, tile=32):
    """Equal-area tile MSE reductions on tile-divisible inputs."""
    if coarse.shape != refined.shape or coarse.shape != target.shape:
        raise ValueError("Gain input shapes differ.")
    if coarse.shape[-2] % tile or coarse.shape[-1] % tile:
        raise ValueError("Gain labels require complete output tiles.")
    pixel = ((coarse - target).square()
             - (refined - target).square()).mean(1, keepdim=True)
    return F.avg_pool2d(pixel, kernel_size=tile,
                        stride=tile).flatten(1)
```

Listing 2: Gate learning with frozen predictions and labels.

```python
def gate_training_loss(light_and_features, gate, expert,
                       frame0, frame1, target, tile=32, halo=8):
    """Only Gate parameters are trainable; Light and Expert are eval()."""
    with torch.no_grad():
        coarse, features = light_and_features(frame0, frame1)
        n = (coarse.shape[-2] // tile) * (coarse.shape[-1] // tile)
        all_tiles = torch.ones((coarse.shape[0], n),
                               dtype=torch.bool, device=coarse.device)
        refined = refine_selected(expert, frame0, frame1, coarse,
                                  all_tiles, tile, halo)
        gain = signed_gain(coarse, refined, target, tile)
    # Keep the Gate forward pass differentiable.
    score = gate(features.detach())
    return gain_gate_loss(score, gain)
```

F.2 Vectorized composition

The crop batch contains selected windows only. This reference unfolds candidate windows before indexing the chosen crops. Expert batch size therefore scales with selected tiles, while candidate-window storage scales with the full tile grid. Disjoint interiors are reconstructed in row-major tile order and added to the coarse frame before clipping.

Listing 3: Gather, one Expert batch, and disjoint-interior composition.

```python
def refine_selected(expert, frame0, frame1, coarse,
                    selected, tile=32, halo=8):
    """Unfold windows, batch selected crops, scatter interiors."""
    if frame0.ndim != 4 or frame0.shape[1] != 3:
        raise ValueError("Expected [B,3,H,W] frames.")
    if frame0.shape != frame1.shape or frame0.shape != coarse.shape:
        raise ValueError("Frame shapes differ.")
    if tile <= 0 or halo < 0:
        raise ValueError("Invalid tile or halo.")
    b, _, h, w = coarse.shape
    gh, gw = (h + tile - 1) // tile, (w + tile - 1) // tile
    mask = selected.reshape(b, -1).to(device=coarse.device,
                                      dtype=torch.bool)
    if mask.shape[1] != gh * gw:
        raise ValueError("Selection size differs from tile grid.")
    ids = mask.flatten().nonzero(as_tuple=False).flatten()
    if ids.numel() == 0:
        return coarse.clone()
    x = torch.cat((frame0, frame1, coarse), dim=1)
    x = F.pad(x, (halo, gw * tile - w + halo,
                  halo, gh * tile - h + halo), mode="replicate")
    side = tile + 2 * halo
    windows = F.unfold(x, kernel_size=side, stride=tile)
    windows = windows.transpose(1, 2).reshape(b * gh * gw, 9,
                                              side, side)
    pred = expert(windows.index_select(0, ids))
    if pred.shape != (ids.numel(), 3, side, side):
        raise ValueError("Expert must return [K,3,side,side] residuals.")
    interior = pred[..., halo:halo + tile, halo:halo + tile]
    packed = coarse.new_zeros((b * gh * gw, 3, tile, tile))
    packed.index_copy_(0, ids, interior)
    residual = packed.reshape(b, gh, gw, 3, tile, tile)
    residual = residual.permute(0, 3, 1, 4, 2, 5).reshape(
        b, 3, gh * tile, gw * tile)
    return (coarse + residual[..., :h, :w]).clamp(0, 1)
```

F.3 Regression and within-image ranking

Scores and targets are flattened to $[B, N]$ before forming differences. Pairwise comparisons remain within each image. If no target difference exceeds the margin, a differentiable zero is used for the ranking contribution.

Listing 4: Signed-gain regression and valid-pair ranking.

```python
def gain_gate_loss(score, gain, scale=1000.0,
                   rank_weight=0.1, margin=1e-7):
    score, gain = score.flatten(1), gain.detach().flatten(1)
    if score.shape != gain.shape:
        raise ValueError("Scores and gains must match.")
    reg = F.smooth_l1_loss(score, scale * gain)
    ds = score.unsqueeze(2) - score.unsqueeze(1)
    dg = gain.unsqueeze(2) - gain.unsqueeze(1)
    valid = dg.abs() > margin
    rank = (F.softplus(-dg.sign()[valid] * ds[valid]).mean()
            if valid.any() else score.sum() * 0.0)
    return reg + rank_weight * rank
```

G Reproduction Checklist

Table 11: Frozen settings associated with the reported staged checkpoint and evaluation.

Item	Setting
Light / Expert base width	16 / 32
Tile interior / halo	32×32 / 8 pixels
Training crop / batch	128×128 / 16
Staged training epochs	3
Optimizer / initial learning rate	Adam / 10^{-4}
Checkpoint seed	2028
CPU	Intel Core i5-13500H
Python / PyTorch	3.10.11 / 2.12.0+cpu
CPU backends	MKL; oneDNN (MKL-DNN) enabled
CPU input / batch	256×448 / 1
PyTorch CPU threads	8
Warm-up / timed iterations	50 / 200
Independent timing processes	6 (two rounds)
Development examples / tile interventions	500 / 56,000
Locked manifest / available examples	3,282 / 3,281