QUASAR: QUERY-ADAPTIVE SPARSE ATTENTION WITH RECOVERY FOR VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long spatiotemporal attention in video Diffusion Transformers becomes an inference bottleneck as sequences grow. Existing sparse attention methods can be tightly coupled to sparse-operator implementations, while accuracy compensation may be insufficient or overly complex, limiting the speed-accuracy trade-off of sparse attention. We propose QUASAR, which uses a block-relevance score to select KV blocks for exact computation and restores the contributions of the remaining blocks. Selection has two variants: Full-Domain scores all legal KV blocks, while Candidate-Constrained applies the same score within a candidate set provided by an external sparse-attention method. Recovery estimates the contributions of unselected blocks from Value-block summaries and combines them with exact attention under the same normalization, with Recovery states at multiple query granularities. We prove output-error bounds for both Candidate-Constrained and Recovery. In Wan2.1, QUASAR with per-query Recovery at 85% sparsity achieves VBench AVG5 of 0.8301 (Dense: 0.8347) with $1.54\times$ end-to-end speedup. At 720p, QUASAR reaches up to $2.41\times$ end-to-end speedup with more than 93% sparsity and remains about 0.005 VBench AVG5 drop relative to dense attention. Results across sparse-attention backends, video models, KV-block geometries, and hardware demonstrate QUASAR's portability.

1 INTRODUCTION

Diffusion Transformer-based video generation models, e.g. DiT Peebles & Xie (2023), Wan Wan et al. (2025), and HunyuanVideo Kong et al. (2024), increasingly operate on longer videos and higher resolutions. Hence, the spatiotemporal sequences grow substantially, making attention blocks the major computation and memory bottleneck for inference. Sparse attention alleviates this overhead by reducing the exact computation to selected key-value (KV) blocks, as demonstrated by VSA Zhang et al. (2025) and Sparse Video-Gen Xi et al. (2025). Sparse attention selects the KV blocks most relevant to the current queries. The importance of each block is estimated from compressed Query/Key representations or spatiotemporal structure Zhang et al. (2025); Shen et al. (2025); Xi et al. (2025), or the scoring cost is reduced through methods such as antidiagonal sampling Xu et al. (2025); exact attention is then computed on the highest-ranked blocks.

However, these methods face two challenges. First, *selection and execution are tightly coupled*: relevance estimation, candidate generation, and block-sparse operators are often designed together. Any change in the candidate source, backend, or hardware platform may require reimplementation of both selection and execution, while the scoring cost grows with the full KV range, making replacement and reuse difficult. Second, *compensation for unselected blocks can be insufficient or computationally expensive*. Directly discarding such blocks removes their contributions to both the attention numerator and normalization, which can distort the attention output and degrade generation quality. Other mechanisms, such as Taylor expansions, score-based compensation, and error-aware routing Li et al. (2026b); Zhou et al. (2026); Li et al. (2026a), incur additional computational cost; their overhead can dilute speedups at lower sparsity, while their effectiveness can degrade as sparsity increases.

To address both challenges, we propose QUASAR (Query-Adaptive Sparse Attention with Recovery). For block selection, QUASAR relies on a mean-pooled block-relevance score and a Top-K rule, applied either to all legal KV blocks (Full-Domain) or within the candidate set provided by

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.

an external sparse-attention method (Candidate-Constrained). The selected blocks are passed to the chosen sparse-attention backend for exact computation. For compensation, Recovery evaluates unselected-block contributions from Value-block summaries and combines them with exact attention under the same normalization. The Recovery state can be shared within query groups of different sizes, from one state per query block to one per query. To allow a flexible trade-off between computational cost and recovery accuracy, we provide output-error bounds for each granularity.

We evaluated QUASAR through extensive experiments on Wan2.1 Wan et al. (2025) at 480p and 720p across sparse-attention backends. For 480p videos, QUASAR Full-Domain per-query Recovery achieves up to $1.54\times$ speedup over the FlashAttention-3 (FA3) Dense baseline with VBench AVG5 Huang et al. (2024a) of 0.8301 versus 0.8347 for Dense. At 720p, QUASAR reaches $2.41\times$ end-to-end speedup over the FA3 Dense baseline with AVG5 0.8215, an AVG5 drop of about 0.005 relative to matched Dense, and is faster than every evaluated sparse baseline. These results transfer across sparse-attention backends, video models, KV-block geometries, and hardware.

The main contributions of this paper are as follows:

- **Unified selection rule works across candidate sources and backends.** QUASAR applies the same block-scoring and Top-K selection rule either over all legal KV blocks (Full-Domain) or within an externally supplied candidate set (Candidate-Constrained), allowing candidate generators and execution backends to be changed independently.
- **Controllable accuracy impact for external candidates.** Under a coverage condition on the externally supplied candidate set, Candidate-Constrained and Full-Domain select the same blocks; when coverage is partial, we give an output-error bound for the resulting selection change. This bounds the attention error of reusing external candidate methods.
- **Reliable Recovery at multiple granularities.** Recovery can use one state shared across a query block, share states within groups, or use an independent state for each query. This multi-granularity strategy allows for choices between recovery accuracy and computational cost, and output-error bounds at each granularity are also proved. The experiments show a gain of AVG5 up to $+0.0344$ with less than 2% E2E overhead by recovery.
- **Portable validation across models, backends, and hardware.** We validate QUASAR across sparse-attention backends, video models, KV-block geometries, and hardware. In Wan2.1, E2E speedup reaches $2.41\times$ at 720p and $1.54\times$ at 480p, with an AVG5 drop of about 0.005 relative to Dense Attention.

2 Related work

Query-adaptive sparse selection. Video sparse attention estimates which regions merit exact computation. VSA uses pooled representations Zhang et al. (2025), DraftAttention uses low-resolution Q/K guidance Shen et al. (2025), and XAttention samples antidiagonals Xu et al. (2025). Sparse VideoGen2 uses semantic clustering Yang et al. (2025), while SPADE uses 3D blocking and input-adaptive estimation Liu et al. (2026a). Recent approaches refine selection through hierarchical scoring and token reordering (DFSAttn Hu et al. (2026)), denoising-trajectory mask prediction (MOD-DiT Liu et al. (2026b)), distilled tile scores (Veda Han et al. (2026)), or learned routing (VORTA Sun et al. (2025a)). QUASAR provides a training-free selection and recovery interface that reuses candidate indices across exact-attention executors. This deployment interface also separates its contribution from caching or token reduction Liu et al. (2024); Sun et al. (2025b); Peruzzo et al. (2025).

Approximating unselected attention. PISA approximates the unselected attention tail with block-wise Taylor terms under common normalization Li et al. (2026b); SVG-EAR combines centroid compensation with error-aware routing Zhou et al. (2026). PASA combines dynamic budgeting, grouped approximation, and stochastic selection Zhang et al. (2026). QUASAR uses key/value-block summaries with query-block, grouped, or per-query weights, and merges the resulting numerator and denominator with exact attention. Its granularity parameter controls query sharing while retaining the same selected blocks and state interface.

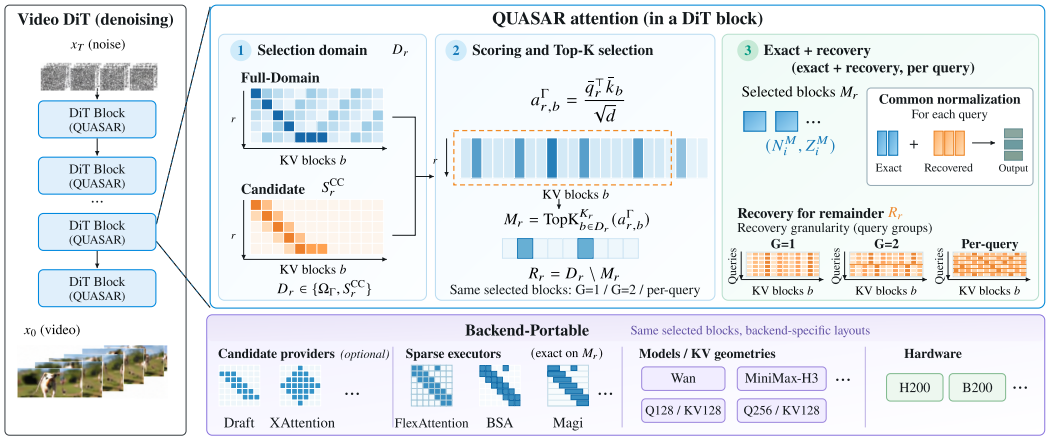


Figure 1: **Overall framework of QUASAR.** The left shows the inputs and Q/K/V projections; the center shows the KV-block set considered for selection, the blocks selected for exact computation, and the recovery branch; the right shows state merging and the output.

Sparse-attention execution. FlashAttention introduced the IO-aware tiling and online-softmax primitives used by dense and sparse attention Dao et al. (2022), with FlashAttention-2 improving parallelism and work partitioning Dao (2024) and FlashAttention-3 adding asynchrony and low precision Shah et al. (2024). Recent video-generation methods combine their selection or approximation policies with specialized block-sparse implementations, including Sparse VideoGen2 Yang et al. (2025), SPADE Liu et al. (2026a), and Sol-Attn Li et al. (2026a). HVA improves clustering cost and tile utilization Lee et al. (2026). QUASAR targets reuse across backends: the same logical block selection feeds backend-specific execution, while recovery joins the online-softmax state.

3 Method

3.1 Problem formulation and block-selection overview

For each attention head, with its index suppressed, G_r denotes a query block and B_b a KV block under geometry Γ. Let Ω_Γ contain the mask-valid KV-block indices. QUASAR operates on a selection domain D_r: Full-Domain uses $D_r = \Omega_\Gamma$, and Candidate-Constrained uses an externally supplied $D_r = S_r^{\mathrm{CC}} \subseteq \Omega_\Gamma$. It partitions this domain into the exact-computation set M_r and recovery set R_r; together they define the Recovery branch (Figure 1).

Both configurations use mean-pooled queries $\bar{q}_r = |G_r|^{-1} \sum_{i \in G_r} q_i$ and keys $\bar{k}_b = |B_b|^{-1} \sum_{j \in B_b} k_j$ to score each KV block B_b:

$$a^\Gamma_{r,b} = \frac{\bar{q}_r^\top \bar{k}_b}{\sqrt{d}}, \quad b \in D_r. \tag{1}$$

For a requested budget K, set $K_r = \min(K, |D_r|)$. Both routes select the highest-scoring K_r blocks:

$$M_r = \mathrm{TopK}^{K_r}_{b \in D_r}(a^\Gamma_{r,b}), \qquad R_r = D_r \setminus M_r. \tag{2}$$

Candidate generators may always include structural blocks, such as Draft's first-frame and same-frame blocks, in D_r; the final Top-K still determines M_r. Blocks outside D_r receive no recovery. Set-equality guarantees use a common tie rule; our numerical reference orders equal scores by ascending logical block ID. Recovery granularity changes neither D_r nor M_r.

Theorem 1 (Candidate coverage). *Let $F_r = M_r^{\mathrm{FD}}$, $e_{r,b} = \widehat{a}_{r,b} - a^\Gamma_{r,b}$, and $\delta_{r,c} = e_{r,c} - \min_{b \in F_r} e_{r,b}$ for $c \notin F_r$, ordered as $\delta_{r,[1]} \geq \delta_{r,[2]} \geq \cdots$. For $1 \leq K \leq L < |\Omega_\Gamma|$, suppose S_r^{CC} contains the proxy Top-L under $\widehat{a}_r$. If, for some $t \in \{0, \ldots, L-K\}$,*

$$a^\Gamma_{r,(K)} - a^\Gamma_{r,(L-t+1)} > \delta_{r,[t+1]}, \tag{3}$$

then $F_r \subseteq S_r^{\mathrm{CC}}$ and $M_r^{\mathrm{CC}} = M_r^{\mathrm{FD}}$ under the common final score and the deterministic tie rule.

Algorithm 1 QUASAR block selection and recovery

Require: $Q, K, V, \Gamma, \{D_r\}, \{K_r\}, G_{\mathrm{rec}}, \beta$
Ensure: $\{\widehat{O}_i\}_i, \{E_{\beta,r}\}_r$
for each query block G_r **do**
  $M_r \leftarrow \mathrm{TopK}^{K_r}_{b\in D_r}(a^{\Gamma}_{r,b})$
  $R_r \leftarrow D_r \setminus M_r$
  $E_{\beta,r} \leftarrow \Lambda_\beta(M_r; \Gamma_\beta)$
  partition G_r into G_{rec} groups
  for each group $G_{r,g}$ **do**
    $(\widehat{Z}^R_{r,g}, \widehat{N}^R_{r,g}) \leftarrow \mathrm{RecoveryState}(R_r, \rho^{\Gamma}_{r,g}, V)$
  end for
  for each query $i \in G_r$ **do**
    $(N^M_i, Z^M_i) \leftarrow \mathrm{ExactState}(i, E_{\beta,r}, Q, K, V)$
    $\widehat{O}_i \leftarrow (N^M_i + \widehat{N}^R_{r,g(i)})/(Z^M_i + \widehat{Z}^R_{r,g(i)})$
  end for
end for
return $\{\widehat{O}_i\}_i, \{E_{\beta,r}\}_r$

The $L-K$ extra slots accommodate t overestimated competitors; coverage depends on the $(t+1)$st largest relative error. Under $|e_{r,b}| \le \epsilon_r$, we have $\delta_{r,[1]} \le 2\epsilon_r$, so $t=0$ recovers the sufficient condition $a^{\Gamma}_{r,(K)} - a^{\Gamma}_{r,(L+1)} > 2\epsilon_r$. Appendix H gives the proof and extensions. If coverage fails, let F and C denote the Full-Domain and Candidate-Constrained exact-computation sets, with normalized exact outputs O^{FD}_i and O^{CC}_i. Their output difference satisfies

$$\eta^{\mathrm{mig}}_i = \max\left\{\frac{W_i(F\setminus C)}{W_i(F)}, \frac{W_i(C\setminus F)}{W_i(C)}\right\}, \qquad \left\|O^{\mathrm{FD}}_i - O^{\mathrm{CC}}_i\right\|_2 \le \eta^{\mathrm{mig}}_i \left\|\mu^+_i - \mu^-_i\right\|_2. \tag{4}$$

Here $W_i(A)$ is the exact attention mass of the block set A; the value centroids $\mu^{\pm}_i$ and the proof are given in Appendix I. The output change is controlled by displaced attention mass and value geometry, and is zero when $F = C$.

3.2 Composable execution and recovery granularity

Backend mapping. Backend β maps the logical exact-computation set to its execution representation, $E_{\beta,r} = \Lambda_\beta(M_r; \Gamma_\beta)$, and returns the exact numerator–denominator state (N^M_i, Z^M_i) for each query. The common scorer and selector consume candidate block indices; the executor consumes M_r; recovery consumes R_r. This separation allows the same selection to run on different backends (Figure 1, center). Appendix F.1 specifies the interface and measured costs.

Recovery granularity. Here G_{rec} denotes the number of query groups used by Recovery. The recovery partitions the query block but keeps the same recovery set R_r for all groups:

$$G_r = \dot{\bigcup_{g=1}^{G_{\mathrm{rec}}}} G_{r,g}, \qquad \bar{q}_{r,g} = \frac{1}{|G_{r,g}|}\sum_{i\in G_{r,g}} q_i, \qquad \rho^{\Gamma}_{r,g,b} = \frac{\bar{q}^{\top}_{r,g}\bar{k}_b}{\sqrt{d}}, \quad b \in R_r. \tag{5}$$

Let $\mathcal{V}_r$ be the valid KV-token set shared by G_r, and define $n_b = |B_b \cap \mathcal{V}_r|$, $V^{\Sigma}_b = \sum_{j\in B_b\cap\mathcal{V}_r} v_j$, and $R^+_r = \{b \in R_r : n_b > 0\}$. The recovery state at group g is

$$\widehat{Z}^R_{r,g} = \sum_{b\in R^+_r} n_b e^{\rho^{\Gamma}_{r,g,b}}, \qquad \widehat{N}^R_{r,g} = \sum_{b\in R^+_r} e^{\rho^{\Gamma}_{r,g,b}} V^{\Sigma}_b. \tag{6}$$

The same pair $(\widehat{N}^R_{r,g}, \widehat{Z}^R_{r,g})$ is reused for every query $i \in G_{r,g}$. The exact and recovery states use common normalization:

$$\widehat{O}_i = \frac{N^M_i + \widehat{N}^R_{r,g(i)}}{Z^M_i + \widehat{Z}^R_{r,g(i)}}. \tag{7}$$

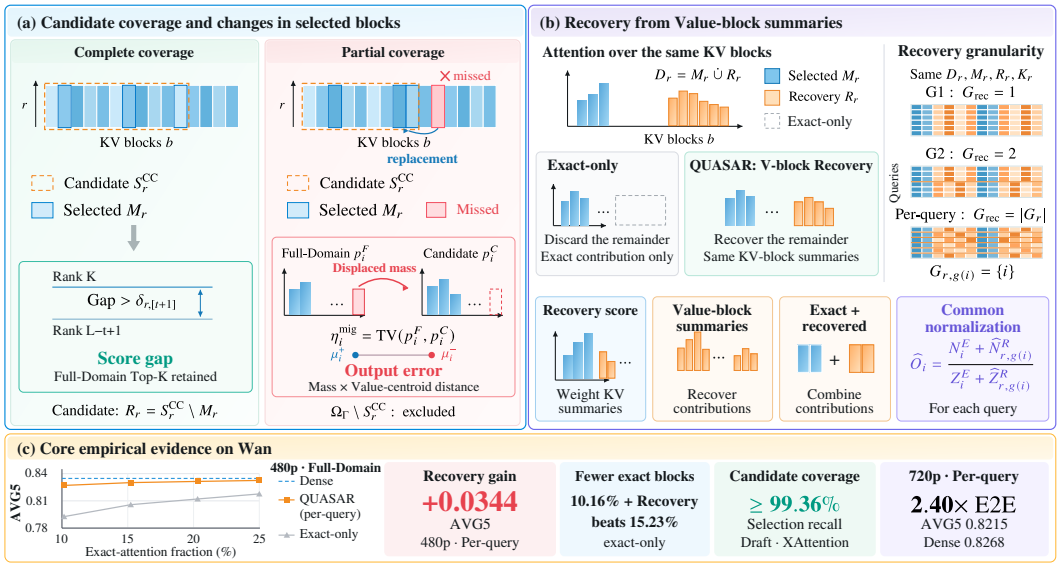


Figure 2: **Candidate coverage, changes in the selected set, and recovery from value-block summaries.** (a) A candidate KV-block set may contain or miss the Full-Domain Top-K blocks. The coverage condition accommodates up to $t \le L - K$ overestimated competitors (Theorem 1). (b) Recovery approximates the unselected blocks and merges them with the exact result. (c) Quality and latency under different numbers of exactly computed blocks.

Per-query recovery. One or two groups share recovery states across a query block; $G_{\text{rec}} = |G_r|$ gives each query its own weights over the same summaries. For nested partitions, finer grouping reduces the aggregate squared block-logit error due to query sharing, and per-query recovery removes that component (Appendix F.2). KV-block refinement addresses the remaining variation within the block. Algorithm 1 summarizes the common computation.

3.3 Attention-Output Error Bounds

For a query $i \in G_{r,g(i)}$, the exact output over the selection domain $D_r = M_r \dot\cup R_r$ is the mixture

$$O_i = (1 - \alpha_i) O_i^M + \alpha_i O_i^R, \qquad \alpha_i = \frac{Z_i^R}{Z_i^M + Z_i^R}, \tag{8}$$

while Eq. (7) uses the recovered branch weight

$$\widehat{\alpha}_{i,g} = \frac{\widehat{Z}_{r,g}^R}{Z_i^M + \widehat{Z}_{r,g}^R}. \tag{9}$$

The total variation (TV) terms Gibbs & Su (2002) measure within-block deviation from uniform token weights and mismatch between exact and recovered block masses; the value diameters scale their output impact. Appendix K.1 collects these quantities and boundary conventions.

Theorem 2 (Hierarchical output error for recovery from value-block summaries)**.** *For every query with positive exact and recovered normalization, the output error decomposes as*

$$\widehat{O}_i - O_i = \Delta_i^{\text{within}} + \Delta_i^{\text{between}} + \Delta_i^{\text{branch}}, \tag{10}$$

and obeys the single hierarchical bound

$$\begin{aligned} \|\widehat{O}_i - O_i\|_2 &\le B_i^{\text{within}} + B_i^{\text{between}} + B_i^{\text{branch}} \\ &= \widehat{\alpha}_{i,g(i)} \left[\sum_{b \in R_r^+} \beta_{i,b} \tau_{i,b}^{\text{tok}} D_b^V + \tau_{i,g(i)}^{\text{blk}} D_R^{\bar{V}} \right] + |\widehat{\alpha}_{i,g(i)} - \alpha_i| \Delta_i^{RM}. \end{aligned} \tag{11}$$

Table 1: **Quality, sparsity, and latency on Wan.** Both Full-Domain and Candidate-Constrained use per-query recovery. "DG=5" applies dense attention only to the first five denoising steps (Dense Guard = 5). Sparsity follows each method's block-selection rule, and E2E includes any dense prefix. Bold font is used for the best speedup and the least latency.

Method	Sparsity	AVG5	Sub.	BG	Mot.	Aes.	Img.	Dyn.	E2E(s)	Speedup
480p (480×832)										
Dense	0.00%	0.8347	0.9753	0.9703	0.9921	0.5627	0.6732	0.3438	272.657	1.00×
DraftAttention	75.00%	0.8207	0.9577	0.9627	0.9899	0.5496	0.6437	0.4271	241.722	1.13×
Sol-Attn ($\tau = 2$)	88.52%	0.8343	0.9720	0.9681	0.9922	0.5615	0.6778	0.3542	200.498	1.36×
XAttention	75.00%	0.8169	0.9497	0.9591	0.9901	0.5479	0.6373	0.4167	204.345	1.33×
PISA	85.16%	0.8348	0.9725	0.9684	0.9924	0.5640	0.6768	0.3438	200.076	1.36×
SVG2	89.36%	0.8332	0.9741	0.9688	0.9920	0.5621	0.6690	0.3333	283.679	0.96×
QUASAR Full-Domain with per-query recovery										
QUASAR-Full	79.69%	0.8314	0.9651	0.9665	0.9916	0.5526	0.6813	0.4167	184.304	1.48×
QUASAR-Full	84.77%	0.8301	0.9613	0.9626	0.9915	0.5492	0.6858	0.3958	**177.155**	**1.54**×
QUASAR-Full	75.00%	**0.8325**	0.9675	0.9670	0.9916	0.5551	0.6812	0.4167	191.568	1.42×
QUASAR Candidate-Constrained										
QUASAR-Draft	79.69%	0.8249	0.9539	0.9604	0.9904	0.5492	0.6704	0.4583	187.234	1.46×
QUASAR-XAttention	79.69%	0.8264	0.9605	0.9610	0.9911	0.5498	0.6694	0.4375	195.215	1.40×
720p (720×1280)										
Dense	0.00%	0.8268	0.9517	0.9592	0.9875	0.5620	0.6738	0.5104	1031.087	1.00×
DraftAttention	75.00%	0.8009	0.8985	0.9366	0.9806	0.5423	0.6462	0.6875	664.637	1.55×
XAttention	80.03%	0.7957	0.8940	0.9358	0.9806	0.5339	0.6342	0.6563	562.209	1.83×
Sol-Attn ($\tau = 2$)	88.52%	0.8272	0.9486	0.9614	0.9884	0.5585	0.6789	0.5313	584.440	1.76×
PISA	85.16%	0.8279	0.9488	0.9594	0.9880	0.5639	0.6791	0.5104	647.646	1.59×
SVG2	89.36%	0.8243	0.9479	0.9584	0.9870	0.5586	0.6696	0.4688	882.280	1.17×
QUASAR Full-Domain with per-query recovery										
QUASAR-Full (DG=0)	93.40%	0.8215	0.9360	0.9573	0.9875	0.5441	0.6825	0.5729	**428.712**	**2.41**×
QUASAR-Full (DG=5)	91.20%	0.8230	0.9449	0.9566	0.9884	0.5533	0.6718	0.5313	506.179	2.04×
QUASAR-Full (DG=0)	89.17%	**0.8247**	0.9340	0.9538	0.9862	0.5522	0.6972	0.5938	465.129	2.22×
QUASAR-XAttention (DG=5)	91.20%	0.8227	0.9428	0.9567	0.9882	0.5561	0.6696	0.5521	519.946	1.98×

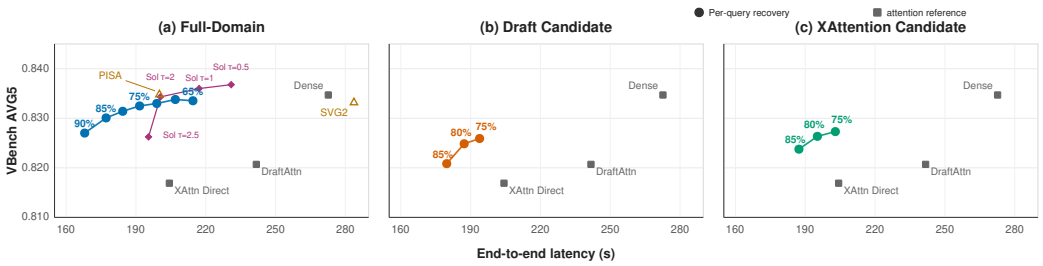


Figure 3: **Pareto results on Wan.** Panels (a)/(b)/(c) show Full-Domain, Draft Candidate, and XAttention Candidate per-query recovery; labels give KV-block sparsity. Gray squares denote Dense, DraftAttention, and XAttention; magenta diamonds denote Sol-Attn thresholds.

The right-hand side defines B_i^{hier}; its three terms correspond to within-block value approximation, between-block mass mismatch, and mismatch between exact and recovery branch weights.

Proof sketch. Expanding the mixture in Eq. (8) around $\bar{v}_b$ gives $\widehat{O}_i - O_i = \Delta_i^{\text{within}} + \Delta_i^{\text{between}} + \Delta_i^{\text{branch}}$. TV and value diameters bound the first two terms; the third satisfies $\|\Delta_i^{\text{branch}}\|_2 \le |\widehat{\alpha}_{i,g(i)} - \alpha_i| \Delta_i^{RM}$. The triangle inequality yields Eq. (11). See Appendix K.4 for the full proof. □

The decomposition identifies two measurable sources of approximation: query-group variation and within-block K/V variation. Appendix K.13 turns query deviations and key/value radii into an input-dependent error bound without enumerating token-pair logits; per-query recovery removes its query-sharing term.

Appendix K.6 also derives a spectral error bound from the Hammersley–Chapman–Robbins (HCR) inequality Nishiyama (2019), together with its extensions to candidate-set selection and error relative to dense attention.

Table 2: **Candidate coverage and matched random controls.** Draft varies the sparsity at a fixed mean candidate fraction. XAttention and Random vary the candidate fraction at the same 79.69% sparsity. Selection recall is evaluated on captured Q/K/V tensors.

Candidate source	Candidate frac.	Sparsity↑	Selection recall↑	AVG5↑	E2E (s)↓
Draft	53.38%	84.77%	99.964%	0.8208	**179.731**
Draft	53.38%	79.69%	99.820%	0.8249	187.234
Draft	53.38%	75.00%	99.363%	0.8259	193.862
XAttention	30.47%	79.69%	99.992%	0.8173	200.631
XAttention	40.63%	79.69%	100.000%	0.8204	200.749
XAttention	60.94%	79.69%	100.000%	**0.8264**	195.215
Random	30.47%	79.69%	30.389%	0.7312	183.836
Random	40.63%	79.69%	40.554%	0.7799	184.016
Random	60.94%	79.69%	60.956%	0.8091	184.312

Table 3: **Recovery comparisons on Wan 480p.** `Base` denotes execution without Recovery; Per-q. uses per-query Recovery at the same block selection.

Selection	Sparsity	exact-only AVG5	Per-q. AVG5	Δ	exact-only E2E(s)	Per-q. E2E(s)	Speedup
Full-Domain	89.84%	0.7926	0.8270	+0.0344	**167.832**	**167.957**	**1.62**×
Draft Candidate	89.84%	0.7803	0.8144	+0.0341	172.087	173.103	1.58×
XAttention Candidate	89.84%	0.7921	0.8180	+0.0259	185.114	185.697	1.47×
Full-Domain	84.77%	0.8058	**0.8301**	+0.0243	174.524	177.155	1.54×
Draft Candidate	84.77%	0.8021	0.8208	+0.0187	179.061	179.731	1.52×
XAttention Candidate	84.77%	0.8053	0.8238	+0.0185	193.164	187.274	1.46×

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and runtime environment. We conduct the main experiments on Wan2.1 Wan et al. (2025) using the LightX2V LightX2V Contributors (2025) inference framework on NVIDIA H200 GPUs, generating 81-frame videos at 480×832 (480p) and 720×1280 (720p) with 50 denoising steps. Paired comparisons share the checkpoint, generation configuration, prompts, and seed. Sparse execution uses FlexAttention Dong et al. (2025), Block Sparse Attention (BSA) Guo et al. (2024), or Magi Tao & Huang (2025) with Q128/KV128 blocks.

Attention and candidate configurations. We compare FA3 Dense, existing sparse methods, and QUASAR with Full-Domain, Draft, or XAttention selection. The 480p main QUASAR configurations span 75.00%–84.77% exact-block sparsity; controlled comparisons vary candidate coverage and recovery granularity. The 720p sweep also includes five initial dense denoising steps. The H200 generation comparison uses FA3 for Dense; same-backend operator controls isolate the effect of sparse execution (Appendix Table 18).

Quality and latency evaluation. We evaluate complete videos with VBench Huang et al. (2024b). AVG5 averages Subject Consistency(Sub.), Background Consistency(BG), Motion Smoothness(Mot.), Aesthetic Quality(Aes.), and normalized Imaging Quality(Img.). Dynamic Degree(Dyn) is reported separately: a higher value means a larger fraction of videos have appreciable motion, and is interpreted alongside smoothness and consistency. E2E uses the LightX2V pipeline interval. Appendix C specifies the prompts, seeds, aggregation, and timing boundaries.

4.2 MAIN RESULTS, COVERAGE, AND RECOVERY

Quality, sparsity, and latency. At 480p, QUASAR-Full reaches 0.8301–0.8325 AVG5 versus Dense 0.8347, reducing E2E latency from 272.657 s to 177.155–191.568 s (up to $1.54\times$; Table 1). At 720p, its fastest configuration achieves $2.41\times$ E2E speedup with 0.8215 AVG5, outperforming every evaluated sparse baseline in latency; The Appendix Table 13 gives the full sweep. Across both resolutions, these configurations remain within 0.007 AVG5 of the matched Dense. Same-backend FlexAttention execution at 79.69% sparsity reduces the fixed-set attention call from 57.738

Table 4: **Cross-model transfer to MiniMax-H3.** The first three rows use matched prompts and seed 43 without recovery; the final row reports an independent Dense evaluation.

Method	Candidate blocks	Selected blocks	Sparsity↑	AVG5↑	E2E (s)↓	Speedup↑
Dense	292/292	292/292	0.00%	0.8352	155.185	1.00×
QUASAR-Full	292/292	59/292	79.79%	0.8306	100.385	**1.55**×
QUASAR-XAttention	89/292	59/292	79.79%	0.8306	106.616	1.46×



Figure 4: **Qualitative comparisons on Wan at 720p.** Rows compare Dense, XAttention, PISA, and QUASAR; left labels report mean E2E generation latency in seconds. The two examples are the spatial-relationship and multiple-objects prompts, with frames 0, 27, 54, and 80 shown in each row. QUASAR uses per-query recovery at 91.20% sparsity with a five-step dense prefix.

to 12.651 ms (4.564×); Magi reaches 11.949 ms (2.537× relative to matched FA3 Dense; Appendix Table 18).

Candidate coverage. XAttention Candidate scores 354 of 591 legal KV blocks at 720p and 156 of 256 at 480p. Table 2 relates this reduction to selection recall and generation quality: Draft retains at least 99.363% of Full-Domain selections; XAttention retains at least 99.992% at candidate fractions of 30.47%, 40.63%, and 60.94%, while matched random candidates retain 30.389%–60.956%. At 79.69% sparsity, AVG5 is 0.8314 for Full-Domain, 0.8264 for XAttention Candidate, and 0.8249 for Draft Candidate, a maximum gap of 0.0065. On MiniMax-H3, both routes report AVG5 0.8306 (Table 4). Candidate size thus trades scoring work against coverage and the amount of recoverable context. Figure 5 relates observed recall to the uniform-error specialization of Theorem 1.

Table 5: **Recovery grouping at 79.69% sparsity.** Rel-L2 fixes captured Wan Q/K/V and exact blocks; AVG5 and E2E use matched generated videos.

G_{rec}	Rel-L2↓	AVG5↑	E2E (s)↓
Exact-only	0.2036	0.8121	182.623
1	0.1890	0.8305	**182.075**
2	0.1886	0.8307	184.005
$\lvert G_r \rvert$	**0.1853**	**0.8314**	184.304

With XAttention candidates, QUASAR improves both quality and E2E speedup over standalone XAttention at their reported sparsities: at 480p, AVG5 and speedup increase from 0.8169 and 1.33× to 0.8264 and 1.40× (Table 1); at 720p, they rise from 0.7957 and 1.83× to 0.8140 and 2.15× (Appendix Table 13). At matched 75.00% sparsity on 480p, QUASAR-XAttention still raises AVG5 to 0.8273 while reducing E2E from 204.345 to 202.928 s (Appendix Table 9). With five initial dense steps at 720p, it reaches a higher-quality point of 0.8227 AVG5 at 1.98× speedup.

4.3 Recovery and Quality–Latency Trade-offs

At 89.84% sparsity, Full-Domain per-query recovery raises AVG5 from 0.7926 without recovery to 0.8270, a gain of 0.0344 (Table 3). The controlled operator ablation fixes captured Q/K/V and exact blocks: Rel-L2 falls from 0.2036 without recovery to 0.1890 with $G_{\text{rec}} = 1$ and 0.1853 with per-query recovery. Separately, matched generation raises AVG5 from 0.8121 to 0.8314 at an additional 1.681 s E2E (Table 5). On captured H3 tensors, per-query recovery reduces attention-output error relative to $G_{\text{rec}} = 1$ recovery (Appendix Table 21). Appendix Table 30 reports dense-reference bounds. Across the three evaluated per-query sparsity levels (84.77%, 79.69%, and 75.00%), Draft

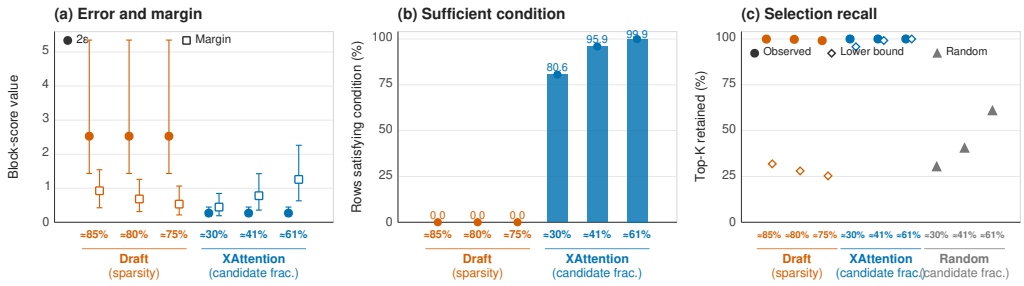


Figure 5: **Candidate-set coverage analysis.** (a) Twice the score error and the Full-Domain rank gap (median and 5th–95th percentiles). (b) Fraction satisfying the uniform-error $2\epsilon_r$ condition. (c) Observed selection recall and its per-block lower bound, with matched Random controls.

Table 6: **Wan transfer on B200.** Dense, online Full-Domain, and online XAttention Candidate results use the same prompt set as previous tables.

Selected blocks	Recovery	Sparsity↑	AVG5↑	Dynamic↑	E2E (s)↓	Speedup↑
256 (Dense)	–	0.0000%	0.835334	0.500000	138.885	1.000×
Full-Domain						
39	Exact-only	84.7656%	0.7902	0.4792	87.8477	1.581×
39	per-query	84.7656%	0.8294	0.5000	**87.1161**	**1.594**×
52	Exact-only	79.6875%	0.8029	0.4896	92.3623	1.504×
52	per-query	79.6875%	0.8293	0.5104	90.5111	1.534×
64	Exact-only	75.0000%	0.8099	0.5000	95.5474	1.454×
64	per-query	75.0000%	**0.8300**	0.5104	93.7560	1.481×
Online XAttention candidate selection: 156 candidate blocks						
52	Exact-only	79.6875%	0.8001	0.4792	104.4595	1.330×
52	per-query	79.6875%	0.8202	0.5000	102.9974	1.348×

Candidate reaches AVG5 0.8208–0.8259 at 179.7–193.9 s, while XAttention Candidate reaches 0.8238–0.8273 at 187.3–202.9 s. Figure 3 shows the broader quality–latency trade-off, and Figure 4 provides 720p comparisons among Dense, XAttention, PISA, and QUASAR.

4.4 Cross-model and cross-deployment transfer

On MiniMax-H3, Full-Domain selection reaches 1.55× E2E speedup and 0.8306 AVG5, within 0.0046 of Dense; XAttention Candidate reaches the same AVG5 at 1.46× (Table 4). Appendix Figure 7 and Table 26 add recovery and seed-42 results. On B200/FA4 with Q256/KV128 geometry, Full-Domain reaches 1.59× speedup at 0.8294 AVG5 and 1.48× at 0.8300; XAttention Candidate reaches 1.35× (Table 6). These transfers reuse the common selection and recovery logic. Appendix F.2 details the integration, and Table 25 reports paired per-query capture measurements.

5 Conclusion

This paper presents QUASAR, which combines KV-block selection, candidate-set coverage analysis, and recovery from key/value-block summaries in a single sparse-attention formulation. The same block-relevance and Top-K rule applies to all mask-valid KV blocks or to an external candidate set; per-query Recovery then estimates the contribution of blocks not computed exactly and merges it with the exact state. The coverage and output-error bounds connect candidate reuse and recovery granularity to approximation accuracy. On Wan2.1, QUASAR attains up to 2.537× fixed-set attention-call speedup over the FA3 Dense operator and 0.8301–0.8325 AVG5 across 75%–85% sparsity, close to Dense 0.8347. Recovery improves the quality–latency trade-off across the evaluated budgets, while MiniMax-H3 and B200 validate transfer across models, geometries, backends, and hardware. At 720p, QUASAR reaches up to 2.41× end-to-end speedup with an AVG5 drop of about 0.005 relative to Dense. QUASAR can be used in the design of a high-performance attention operator for video generation or other computationally heavy tasks.

REFERENCES

Samuel N. Cohen and Eliana Fausti. Hyperbolic contractivity and the hilbert metric on probability measures. *arXiv preprint arXiv:2309.02413*, 2023. doi: 10.48550/arXiv.2309.02413. URL https://arxiv.org/abs/2309.02413.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.08691.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16344–16359, 2022. URL https://arxiv.org/abs/2205.14135.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. FlexAttention: A programming model for generating fused attention variants. In *Proceedings of Machine Learning and Systems*, volume 7, 2025. URL https://proceedings.mlsys.org/paper_files/paper/2025/hash/61a9278dfef5f871b5e472389f8d6fa1-Abstract-Conference.html.

Alison L. Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International Statistical Review*, 70(3):419–435, 2002. doi: 10.1111/j.1751-5823.2002.tb00178.x. URL https://doi.org/10.1111/j.1751-5823.2002.tb00178.x.

Junxian Guo, Haotian Tang, Shang Yang, Zhekai Zhang, Zhijian Liu, and Song Han. Block Sparse Attention. https://github.com/mit-han-lab/Block-Sparse-Attention, 2024.

Shihao Han, Hao Yang, Xinting Hu, Xiaofeng Mei, Yi Jiang, and Xiaojuan Qi. Veda: Scalable Video Diffusion via Distilled Sparse Attention. *arXiv preprint arXiv:2605.30325*, 2026. URL https://arxiv.org/abs/2605.30325.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830. URL https://doi.org/10.1080/01621459.1963.10500830.

Jie Hu, Zixiang Gao, Yutong He, and Kun Yuan. DFSAttn: Dynamic fine-grained sparse attention for efficient video generation. *arXiv preprint arXiv:2605.23445*, 2026.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, 2024a. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_VBench_Comprehensive_Benchmark_Suite_for_Video_Generative_Models_CVPR_2024_paper.html.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, June 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_VBench_Comprehensive_Benchmark_Suite_for_Video_Generative_Models_CVPR_2024_paper.html.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. URL https://arxiv.org/abs/2412.03603.

Dongyeun Lee, Amir Zandieh, Vahab Mirrokni, Junmo Kim, and Insu Han. HyperVAttention: Efficient Sparse Attention with Spatio-Temporal Clustering for Video Diffusion. *arXiv preprint arXiv:2607.03012*, 2026. URL https://arxiv.org/abs/2607.03012.

Haopeng Li, Yitong Li, Junsong Chen, Tian Ye, Haozhe Liu, Jincheng Yu, Duomin Wang, Ruihua Zhang, Zeke Xie, Enze Xie, and Song Han. Sol-Attn: Accelerating video generation inference via on-the-fly attention sparsification. *arXiv preprint arXiv:2607.24027*, 2026a. URL https://arxiv.org/abs/2607.24027.

Haopeng Li, Shitong Shao, Wenliang Zhong, Zikai Zhou, Lichen Bai, Hui Xiong, and Zeke Xie. PISA: Piecewise sparse attention is wiser for efficient diffusion transformers. *arXiv preprint arXiv:2602.01077*, 2026b. URL https://arxiv.org/abs/2602.01077.

LightX2V Contributors. LightX2V: Light video generation inference framework. https://github.com/ModelTC/lightx2v, 2025.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. *arXiv preprint arXiv:2411.19108*, 2024. URL https://arxiv.org/abs/2411.19108.

Shanghao Liu, Renze Chen, Size Zheng, Yuanqiang Liu, Yun Liang, and Hailong Yang. SPADE: An input-adaptive sparse attention engine for fast video diffusion models inference. In *Proceedings of the 63rd ACM/IEEE Design Automation Conference*, 2026a. doi: 10.1145/3770743.3804015. URL https://arxiv.org/abs/2608.03335.

Yuxi Liu, Yipeng Hu, Zekun Zhang, Kunze Jiang, and Kun Yuan. Mixture of Distributions Matters: Dynamic Sparse Attention for Efficient Video Diffusion Transformers. *arXiv preprint arXiv:2601.11641*, 2026b. URL https://arxiv.org/abs/2601.11641.

Tomohiro Nishiyama. A new lower bound for Kullback–Leibler divergence based on Hammersley–Chapman–Robbins bound. *arXiv preprint arXiv:1907.00288*, 2019. doi: 10.48550/arXiv.1907.00288. URL https://arxiv.org/abs/1907.00288.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Elia Peruzzo, Adil Karjauv, Nicu Sebe, Amir Ghodrati, and Amir Habibian. ADAPTOR: Adaptive token reduction for video diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 6419–6425, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025W/EDGE/html/Peruzzo_ADAPTOR_Adaptive_Token_Reduction_for_Video_Diffusion_Transformers_CVPRW_2025_paper.html.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2193. URL https://arxiv.org/abs/2407.08608.

Xuan Shen, Chenxia Han, Yufa Zhou, Yanyue Xie, Yifan Gong, Quanyi Wang, Yiwei Wang, Yanzhi Wang, Pu Zhao, and Jiuxiang Gu. DraftAttention: Fast video diffusion via low-resolution attention guidance. *arXiv preprint arXiv:2505.14708*, 2025. URL https://arxiv.org/abs/2505.14708.

Wenhao Sun, Rong-Cheng Tu, Yifu Ding, Jingyi Liao, Zhao Jin, Shunyu Liu, and Dacheng Tao. Vorta: Efficient video diffusion via routing sparse attention. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-0268.

Wenhao Sun, Rong-Cheng Tu, Jingyi Liao, Zhao Jin, and Dacheng Tao. AsymRnR: Video diffusion transformers acceleration with asymmetric reduction and restoration. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57694–57711. PMLR, 2025b. URL https://proceedings.mlr.press/v267/sun25r.html.

Zewei Tao and Yunpeng Huang. MagiAttention: A distributed attention towards linear scalability for ultra-long context, heterogeneous mask training. https://github.com/SandAI-org/MagiAttention/, 2025.

Alexander Tian, Aditya Ghai, Sanjit Neelam, Zaal Vasania, and Akshay Mishra. COBS: Cumulant order block sparse attention. *arXiv preprint arXiv:2607.09052*, 2026. URL https://arxiv.org/abs/2607.09052.

Georgios Tzachristas, Lei Deng, Ioannis Tzachristas, Gong Zhang, and Renhai Chen. A mathematical theory of top-k sparse attention via total variation distance. *arXiv preprint arXiv:2512.07647*, 2025. doi: 10.48550/arXiv.2512.07647. URL https://arxiv.org/abs/2512.07647.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 68208–68224. PMLR, 2025. URL https://proceedings.mlr.press/v267/xi25c.html.

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. XAttention: Block sparse attention with antidiagonal scoring. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 69819–69831. PMLR, 2025. URL https://proceedings.mlr.press/v267/xu25ag.html.

Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, Jianfei Chen, Song Han, Kurt Keutzer, and Ion Stoica. Sparse VideoGen2: Accelerate video generation with sparse attention via semantic-aware permutation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/8bd5148caced2d73cea7b6961a874a49-Abstract-Conference.html.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric Xing, and Hao Zhang. VSA: Faster video diffusion with trainable sparse attention. *arXiv preprint arXiv:2505.13389*, 2025. URL https://arxiv.org/abs/2505.13389.

Wentai Zhang, Ronghui Xi, Shiyao Peng, Jiayu Huang, Haoran Luo, Zichen Tang, and Haihong E. Ride the Wave: Precision-Allocated Sparse Attention for Smooth Video Generation. *arXiv preprint arXiv:2604.12219*, 2026. URL https://arxiv.org/abs/2604.12219.

Xuanyi Zhou, Qiuyang Mang, Shuo Yang, Haocheng Xi, Jintao Zhang, Huanzhi Mao, Joseph E. Gonzalez, Kurt Keutzer, Ion Stoica, and Alvin Cheung. SVG-EAR: Parameter-free linear compensation for sparse video generation via error-aware routing. *arXiv preprint arXiv:2603.08982*, 2026. URL https://arxiv.org/abs/2603.08982. Accepted to ECCV 2026.

A Reproducibility Statement

All reported generation results use fixed prompts, checkpoints, seeds, denoising schedules, timing boundaries, and native pipeline measurements specified in Appendix C. The main tables report matched prompt aggregates, while the appendix specifies prompt counts, candidate budgets, KV

geometries, recovery granularities, and backend settings. The theory sections state the assumptions and complete proofs for candidate coverage, selected-set migration, and Recovery error, and the numerical validation tables evaluate these bounds on captured Q/K/V tensors. The supplementary material contains the corresponding implementation and experiment scripts. Code are presented in *submission_code.zip*.

B LLM USAGE

Large language models assisted with manuscript revision, including improving clarity, concision, organization, and notation consistency, and with exposition of the theoretical analysis of recovery from value-block summaries. The central research idea, problem formulation, method design, experimental plan, and interpretation of results were developed by the authors. The authors reviewed the resulting manuscript, derivations, citations, implementation details, and experimental results.

C EXPERIMENTAL CONFIGURATION

Models, generation, and prompt sets. Wan experiments use Wan2.1-T2V-14B in BF16, 50 denoising steps, and 81 frames. The 480p main comparison uses height 480 and width 832; the high-resolution evaluation uses 720×1280. The H200 Wan main comparison, candidate coverage tables, and four-threshold Sol comparison use 96 matched prompts and seed 42. The H200 block geometry is Q128/KV128, with 256 legal KV blocks at 480p and 591 legal KV blocks at 720p. The main QUASAR generation results use Magi; the Wan Dense generation reference uses FA3. FlexAttention and BSA additionally evaluate portability of a fixed exact-computation set. B200 experiments use the same Wan generation dimensions, seed 42, eight prompts, and FA4 with Q256/KV128 geometry. Results are aggregated within each hardware and prompt-set combination. The 720p evaluation uses the same 96 matched prompts and seed 42, with 75,600 tokens per runtime sequence.

MiniMax-H3 experiments use the T2VA model in BF16 on H200 with sequence-parallel size four, 50 model evaluations, and 124 frames at 1344×768 resolution. The first three main transfer rows use 96-prompt and seed 43 without recovery; an additional Dense row reports an independent 96-prompt seed-43 evaluation. Supplementary replication uses seed 42. All configurations use Q128/KV128 geometry and 292 legal video KV blocks. Full-Domain uses 79.79% sparsity; XAttention first restricts selection to a 30.48% candidate fraction and uses the same sparsity. Supplementary H3 recovery results use the eight-prompt setting reported with those tables.

Methods and sparsity levels. Full-Domain applies the QUASAR score and selector over all legal KV blocks. Draft Candidate retains DraftAttention's always-included structural blocks and variable-size candidate KV-block set before the common scorer and selector; its reported candidate fraction is averaged over query blocks. XAttention Candidate constructs a sampled candidate KV-block set and applies the same QUASAR scorer and selector. The main Wan XAttention configuration uses a 60.94% candidate fraction; the candidate-size comparison varies this fraction over 30.47%, 40.63%, and 60.94% at a fixed 79.69% sparsity. Uniform sampling supplies the corresponding candidate-set control. DraftAttention and XAttention retain their own selection rules in their reference rows; QUASAR-Draft and QUASAR-XAttention apply QUASAR selection within the corresponding candidate sets. Paired recovery comparisons fix the candidate generator, selection setting, sparsity, backend, prompts, and seed, changing only the recovery computation. The nominal sparsity is the fraction of legal KV blocks omitted; the requested sparsity is a lower bound when a candidate set is smaller. The cross-backend operator comparison uses a fixed exact-computation set; the generation results use the configurations above.

Generation timing. Wan E2E is the arithmetic mean of pipeline latency across prompts. The 720p results use the last native `RUN pipeline cost` entry for each prompt, with the same timing boundary for Dense and sparse methods. The interval includes online candidate construction and exact-block selection, backend conversion, attention, any recovery-state construction and merge, and VAE decoding; MP4 encoding and warm-up are excluded. MiniMax-H3 uses the maximum recorded `RUN pipeline cost` over four ranks for each prompt and then averages across prompts. The main H200 Wan and H3 speedups use their respective matched Dense means, 272.657

Table 7: **Recovery-state granularity on Wan.** The $G_{\mathrm{rec}} = 1$ and $G_{\mathrm{rec}} = 2$ configurations use identical Full-Domain selections with Q128/KV128 geometry. Results use 96 prompts at seed 42; Table 5 adds $G_{\mathrm{rec}} = |G_r|$ at 79.69% sparsity.

Recovery	Sparsity (%)	Sub	BG	Mot	Aes	Img	Dyn	AVG5	E2E (s)
$G_{\mathrm{rec}} = 1$	84.77	0.9604	0.9618	0.9911	0.5406	0.6906	0.4063	0.8289	177.262
$G_{\mathrm{rec}} = 2$	84.77	0.9590	0.9623	0.9914	0.5434	0.6903	0.3958	0.8293	**176.612**
$G_{\mathrm{rec}} = 1$	79.69	0.9658	0.9652	0.9914	0.5440	0.6860	0.3854	0.8305	182.075
$G_{\mathrm{rec}} = 2$	79.69	0.9656	0.9661	0.9914	0.5445	0.6860	0.3854	0.8307	184.005
$G_{\mathrm{rec}} = 1$	75.00	0.9690	0.9661	0.9917	0.5527	0.6823	0.3958	0.8324	191.722
$G_{\mathrm{rec}} = 2$	75.00	0.9692	0.9680	0.9918	0.5516	0.6838	0.3750	**0.8329**	190.782

s and 155.185 s. The Wan sparsity sweep uses its matched FA3 Dense mean of 272.997 s, and the 720p evaluation uses its matched FA3 Dense mean of 1031.087 s.

The B200 table presents Full-Domain and XAttention configurations with matched recovery comparisons using value-block summaries. Each comparison fixes the selection setting and prompt set; all displayed speedups use the Dense attention of $138.885\,\mathrm{s}$.

Operator timing at fixed selected blocks. The H200 operator benchmark uses BF16 Q/K/V of shape $[32760, 40, 128]$, Q128/KV128 geometry, and a 120-token tail block. One captured Wan attention head is repeated across 40 heads for this controlled execution benchmark. For each selection configuration, all backends receive identical inputs, valid-token masks, and the same exact-computation set. Kernel timing starts with prepared layouts and descriptors. Fixed-set attention-call timing starts from a fixed exact-computation set and includes layout preparation, backend mapping, attention execution, and output restoration. Candidate construction, relevance scoring, and Top-K selection are included in the generation timing above.

Table 18 reports CUDA-event medians after five warm-up iterations and 20 rotating measurements. Loading, host-to-device transfer, and compilation are not included in the timed interval. Kernel and call medians are measured independently. The kernel and call speedup columns divide the common Flex Dense measurements, 57.896 ms and 57.738 ms, by each sparse latency. The FA3 comparison instead uses the matched FA3 Dense latency for each execution setting. Relative L2 uses the FA3 Dense output. Backend agreement is evaluated on the same exact-computation set. The grouped recovery operator measurements use three warm-up iterations and 20 rotating measurements at fixed inputs; the per-query comparison in Table 24 uses 30 repeats.

Comparison settings. Sol-Attn uses its SM90 routing, threshold preparation, and correction kernel with $\tau \in \{0.5, 1, 2, 2.5\}$. Its Wan configuration uses dense attention for the first ten of 50 denoising steps and for the first layer of subsequent steps; those calls are included in the reported generation time. Table 20 additionally reports 42 matched prompts. Figure 3 and Table 19 pair quality and timing from the same generated videos.

The four-threshold timing analysis covers the same eight prompts per threshold, with 4,000 self-attention calls per video: 880 dense-attention calls and 3,120 Sol-Attn calls. Valid-token-pair sparsities inside Sol-Attn calls are 66.7422%, 76.4514%, 88.5165%, and 92.3019%; including the dense-attention calls, they are 52.0589%, 59.6321%, 69.0429%, and 71.9955%. The timed videos match the corresponding untimed references, and the additional measurement overhead is excluded from E2E. These eight-prompt statistics describe the Sol-Attn execution; quality aggregates are reported in the main tables.

PISA and SVG2 are included in Table 1 using the same 96-prompt aggregation as the other Wan rows. E2E uses the pipeline interval defined above.

Quality metrics and statistical analysis. AVG5 is the equal-weight mean of Subject Consistency, Background Consistency, Motion Smoothness, Aesthetic Quality, and Imaging Quality, with Imaging Quality normalized to $[0, 1]$. Dynamic Degree is reported independently. Recovery comparisons pair the same prompt and seed before aggregation. The seed-42 confidence intervals use 10,000 suite-stratified paired prompt bootstrap samples, preserving the prompt count in each of the 11 suites (eight prompts per suite), and report percentile 95% intervals. The seed-43 evaluation

independently repeats generation. Candidate score fitting and held-out splits are specified in Appendix D; the attention-error evaluation uses the captured Q/K/V tensors accompanying Table 30.

Additional configurations. At 84.77% sparsity, Full-Domain recovery with $G_{\mathrm{rec}} = 1$ attains AVG5 0.8289 at 177.262 s ($1.54\times$ relative to the matched FA3 Dense mean). At representative 75%–85% sparsity, XAttention's measured E2E overhead is 0.30%–0.37%, and the Full-Domain complete-attention-call overhead is 1.09%–1.82%. Across the same sparsity levels, the independent seed-43 recovery gains span 0.0095–0.0267 for Full-Domain, 0.0085–0.0238 for Draft, and 0.0052–0.0222 for XAttention.

D CANDIDATE COVERAGE BOUND CALIBRATION

Draft uses a head-wise positive affine calibration in log-score space, $g_h(z) = \alpha_h \log z + \beta_h$, fitted to the maximum low-resolution attention over overlapping Q/K buckets. We evaluate heads 5 and 17 using one prompt and seed. For each head, two captured activation tensors fit the map and two are held out, giving four fitting tensors, four held-out tensors, and 1,024 held-out rows per configuration. XAttention fits a positive affine map on 2,048 rows of one captured tensor and evaluates 8,192 disjoint rows. Figure 5(a) reports held-out error medians and p05–p95 intervals; panel (c) reports mean selection recall.

Table 8 reports the uniform-error specialization of Theorem 1 and the per-block recall bound from $\mathcal{A}_r^U$ (Appendix H). "Condition met" is the fraction satisfying the $2\epsilon_r^U$ complete-coverage condition; "Recall LB" averages these per-block lower bounds. Draft's observed selection recall remains above 99.1%.

Table 8: **Held-out uniform-error coverage bounds for Draft.**

Sparsity	Candidate frac.	Rows	Condition met	Recall LB	Selection recall
84.77%	53.38%	1,024	0.000000	0.318785	0.999324
79.69%	53.38%	1,024	0.000000	0.280142	0.997859
75.00%	53.38%	1,024	0.000000	0.253281	0.991089

With a 60.94% candidate fraction, XAttention reaches selection recalls of 99.990%, 99.980%, and 99.965% at 84.77%, 79.69%, and 75.00% sparsity, respectively. The corresponding generation scores and latencies appear in Table 9.

E ADDITIONAL QUALITY–LATENCY RESULTS

The Wan tables below use H200 and seed 42, following Appendix C. AVG5 excludes Dyn and uses Img normalized to $[0, 1]$; E2E is mean pipeline latency in seconds. At 480p, XAttention uses a 60.94% candidate fraction, while Draft's candidate count varies across query blocks.

Per-query recovery. Table 9 reports the complete evaluated sparsity sweep on 96 matched prompts. Full-Domain reaches its highest measured AVG5 at 64.84% sparsity; further increasing the number of exactly computed blocks does not improve AVG5. Both candidate configurations improve quality as more blocks are computed exactly. Dense and sparse-attention baselines are given in Table 1.

Table 9: **Per-query Pareto results on Wan 480p.** Draft and XAttention denote candidate generators.

Selection	Sparsity	Sub	BG	Mot	Aes	Img	Dyn	AVG5	E2E (s)
Full-Domain	89.84%	0.9576	0.9611	0.9913	0.5400	0.6851	0.3750	0.8270	**167.957**
Full-Domain	84.77%	0.9613	0.9626	0.9915	0.5492	0.6858	0.3958	0.8301	177.155
Full-Domain	79.69%	0.9651	0.9665	0.9916	0.5526	0.6813	0.4167	0.8314	184.304
Full-Domain	75.00%	0.9675	0.9670	0.9916	0.5551	0.6812	0.4167	0.8325	191.568
Full-Domain	69.92%	0.9684	0.9669	0.9917	0.5575	0.6804	0.4062	0.8330	198.947
Full-Domain	64.84%	0.9697	0.9685	0.9917	0.5597	0.6793	0.4062	**0.8338**	206.919
Full-Domain	59.77%	0.9674	0.9682	0.9919	0.5598	0.6804	0.4167	0.8335	214.477
Full-Domain	54.69%	0.9652	0.9688	0.9919	0.5566	0.6780	0.3854	0.8321	222.457
Draft	84.77%	0.9487	0.9565	0.9900	0.5423	0.6667	0.4688	0.8208	**179.731**
Draft	79.69%	0.9539	0.9604	0.9904	0.5492	0.6704	0.4583	0.8249	187.234
Draft	75.00%	0.9577	0.9612	0.9906	0.5531	0.6670	0.4479	**0.8259**	193.862
XAttention	84.77%	0.9545	0.9605	0.9909	0.5434	0.6695	0.4583	0.8238	**187.274**
XAttention	79.69%	0.9605	0.9610	0.9911	0.5498	0.6694	0.4375	0.8264	195.215
XAttention	75.00%	0.9623	0.9638	0.9912	0.5541	0.6652	0.4271	**0.8273**	202.928

Recovery across sparsity levels. Figure 6 repeats the recovery comparison with an independent seed. Tables 10–12 extend the $G_{\text{rec}} = 1$ comparison across eight sparsity levels: $G_{\text{rec}} = 1$ improves AVG5 over execution without recovery for every selection method and sparsity shown.

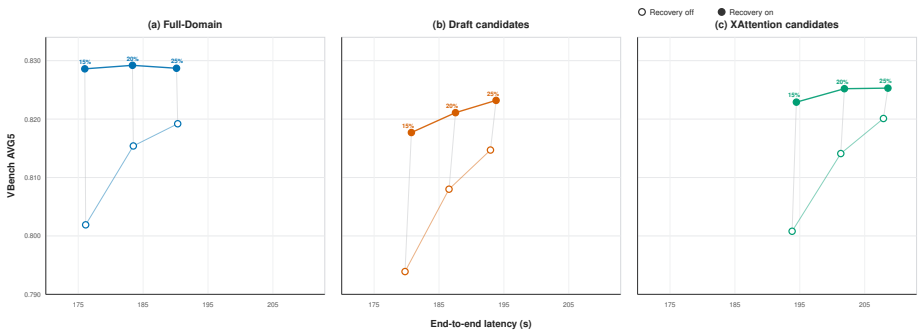


Figure 6: **Independent Wan quality–latency replication.** The three panels compare matched seed-43 exact-only execution and $G_{\text{rec}} = 1$ recovery at 84.77%, 79.69%, and 75.00% sparsity.

Table 10: **Wan 480p sparsity sweep for Full-Domain selection.**

Sparsity (%)	Recovery	Sub	BG	Mot	Aes	Img	Dyn	AVG5	E2E (s)
89.8	Exact-only	0.9219	0.9444	0.9875	0.5158	0.5926	0.4886	0.7924	167.544
89.8	$G_{\text{rec}} = 1$	0.9576	0.9599	0.9909	0.5309	0.6816	0.3295	0.8242	167.424
84.8	Exact-only	0.9355	0.9522	0.9887	0.5319	0.6194	0.5000	0.8056	174.476
84.8	$G_{\text{rec}} = 1$	0.9599	0.9614	0.9912	0.5405	0.6880	0.3977	0.8282	177.458
79.7	Exact-only	0.9439	0.9559	0.9899	0.5437	0.6268	0.4773	0.8120	182.801
79.7	$G_{\text{rec}} = 1$	0.9658	0.9649	0.9916	0.5451	0.6839	0.3750	0.8303	182.040
75.0	Exact-only	0.9526	0.9597	0.9902	0.5520	0.6346	0.4432	0.8178	188.506
75.0	$G_{\text{rec}} = 1$	0.9688	0.9659	0.9918	0.5535	0.6810	0.3864	0.8322	191.944
69.9	Exact-only	0.9587	0.9618	0.9908	0.5513	0.6426	0.4773	0.8211	196.927
69.9	$G_{\text{rec}} = 1$	0.9698	0.9677	0.9919	0.5564	0.6801	0.3750	0.8332	196.810
64.8	Exact-only	0.9629	0.9650	0.9910	0.5613	0.6399	0.4318	0.8240	204.164
64.8	$G_{\text{rec}} = 1$	0.9691	0.9689	0.9917	0.5563	0.6752	0.3864	0.8322	204.097
59.8	Exact-only	0.9655	0.9652	0.9915	0.5604	0.6427	0.3977	0.8250	211.614
59.8	$G_{\text{rec}} = 1$	0.9701	0.9685	0.9916	0.5610	0.6723	0.3864	0.8327	211.480
54.7	Exact-only	0.9657	0.9669	0.9915	0.5570	0.6448	0.4091	0.8252	218.981
54.7	$G_{\text{rec}} = 1$	0.9672	0.9682	0.9918	0.5589	0.6705	0.3750	0.8313	218.811

Candidate-based recovery. The gain persists when scoring is restricted to external candidates. At 89.8% sparsity, Draft's AVG5 rises from 0.7803 to 0.8144, and XAttention's from 0.7921 to 0.8180, with less than 1.1 s additional E2E latency in either comparison.

Table 11: **Wan 480p sparsity sweep with Draft Candidate selection.**

Sparsity (%)	Recovery	Sub	BG	Mot	Aes	Img	Dyn	AVG5	E2E (s)
89.8	Exact-only	0.9064	0.9418	0.9846	0.5010	0.5679	0.4886	0.7803	**172.087**
89.8	$G_{\text{rec}} = 1$	0.9450	0.9547	0.9893	0.5235	0.6595	0.4091	0.8144	173.103
84.8	Exact-only	0.9309	0.9503	0.9879	0.5294	0.6131	0.5114	0.8023	179.088
84.8	$G_{\text{rec}} = 1$	0.9483	0.9561	0.9896	0.5387	0.6681	0.4318	0.8202	179.493
79.7	Exact-only	0.9395	0.9527	0.9884	0.5378	0.6293	0.4886	0.8095	185.206
79.7	$G_{\text{rec}} = 1$	0.9533	0.9592	0.9900	0.5471	0.6730	0.4659	0.8245	188.928
75.0	Exact-only	0.9451	0.9569	0.9894	0.5453	0.6318	0.5114	0.8137	192.211
75.0	$G_{\text{rec}} = 1$	0.9573	0.9621	0.9906	0.5466	0.6668	0.4091	0.8247	192.529
69.9	Exact-only	0.9509	0.9575	0.9899	0.5476	0.6392	0.4432	0.8170	199.065
69.9	$G_{\text{rec}} = 1$	0.9619	0.9627	0.9907	0.5534	0.6641	0.4318	0.8265	199.981
64.8	Exact-only	0.9558	0.9614	0.9906	0.5504	0.6397	0.4545	0.8196	206.784
64.8	$G_{\text{rec}} = 1$	0.9632	0.9644	0.9911	0.5532	0.6623	0.3977	0.8268	207.653
59.8	Exact-only	0.9590	0.9618	0.9909	0.5544	0.6379	0.4318	0.8208	213.725
59.8	$G_{\text{rec}} = 1$	0.9648	0.9662	0.9913	0.5522	0.6569	0.4091	0.8263	214.477
54.7	Exact-only	0.9620	0.9659	0.9910	0.5560	0.6471	0.4091	0.8244	220.694
54.7	$G_{\text{rec}} = 1$	0.9668	0.9655	0.9915	0.5538	0.6572	0.3864	**0.8270**	221.347

Table 12: **Wan 480p sparsity sweep with XAttention Candidate selection.**

Sparsity (%)	Recovery	Sub	BG	Mot	Aes	Img	Dyn	AVG5	E2E (s)
89.8	Exact-only	0.9212	0.9434	0.9875	0.5145	0.5939	0.5000	0.7921	**185.114**
89.8	$G_{\text{rec}} = 1$	0.9460	0.9545	0.9900	0.5312	0.6683	0.3977	0.8180	185.697
84.8	Exact-only	0.9353	0.9510	0.9885	0.5321	0.6195	0.4773	0.8053	193.164
84.8	$G_{\text{rec}} = 1$	0.9535	0.9573	0.9906	0.5409	0.6691	0.4318	0.8223	193.886
79.7	Exact-only	0.9460	0.9565	0.9897	0.5436	0.6302	0.4773	0.8132	200.532
79.7	$G_{\text{rec}} = 1$	0.9584	0.9619	0.9909	0.5489	0.6607	0.4318	0.8242	201.134
75.0	Exact-only	0.9512	0.9591	0.9902	0.5507	0.6362	0.4432	0.8175	207.023
75.0	$G_{\text{rec}} = 1$	0.9619	0.9644	0.9913	0.5469	0.6604	0.4091	0.8250	207.720
69.9	Exact-only	0.9584	0.9605	0.9908	0.5532	0.6406	0.4659	0.8207	214.506
69.9	$G_{\text{rec}} = 1$	0.9643	0.9656	0.9913	0.5549	0.6649	0.4091	0.8282	215.197
64.8	Exact-only	0.9635	0.9644	0.9909	0.5584	0.6384	0.4432	0.8231	221.877
64.8	$G_{\text{rec}} = 1$	0.9648	0.9651	0.9915	0.5576	0.6572	0.4205	0.8273	222.555
59.8	Exact-only	0.9662	0.9650	0.9915	0.5578	0.6453	0.3864	0.8252	229.261
59.8	$G_{\text{rec}} = 1$	0.9655	0.9665	0.9916	0.5535	0.6607	0.3977	**0.8276**	229.907
54.7	Exact-only	0.9660	0.9661	0.9916	0.5572	0.6432	0.3977	0.8248	236.948
54.7	$G_{\text{rec}} = 1$	0.9653	0.9663	0.9918	0.5521	0.6545	0.3523	0.8260	237.651

Cross-model recovery. Figure 7 extends the quality–latency analysis to MiniMax-H3, complementing the transfer results in Table 4. The paired panels compare each selection method with and without recovery at three sparsity levels; the wider Full-Domain sweep covers additional settings.

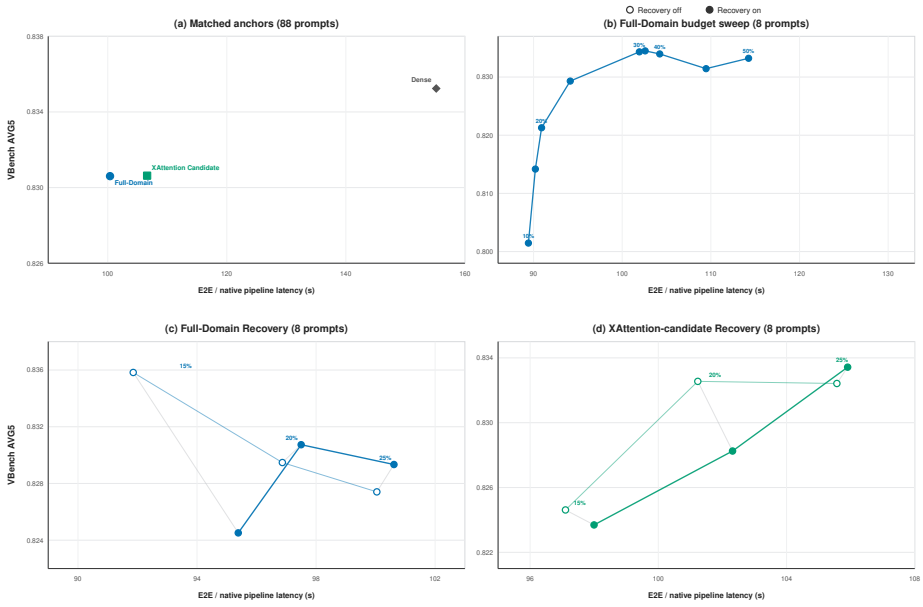


Figure 7: **Quality–latency results on MiniMax-H3.** (a) Seed-43 matched Dense and sparse anchors. (b) Full-Domain sparsity sweep. (c) Full-Domain recovery pairs. (d) XAttention-candidate recovery pairs; panels (b)–(d) use eight prompts.

High-resolution generation. Table 13 reports the 720p evaluation on 96 matched prompts. Per-query Recovery improves Full-Domain AVG5 over $G_{\mathrm{rec}} = 1$ at all three shared sparsities. The runtime sequence has 75,600 tokens and 591 KV blocks of size 128; XAttention uses 354 candidate blocks. The Dense Guard=0 rerun reproduces the previously scored 96-video outputs byte-for-byte, so its updated native E2E timings are directly comparable. In panel (b), Draft's parenthesized value is its candidate threshold. "Dense Guard=5" denotes five initial FA3 dense steps, whose cost is included in E2E. Speedup uses the matched Dense latency of 1031.087 s; baseline sparsities follow each method's selection rule.

Table 13: **Wan 720p quality and latency.** (a) Full-Domain Recovery, (b) Candidate-Constrained per-query Recovery, and (c) matched dense and sparse baselines.

(a) QUASAR 720p sweep: Full-Domain recovery										
Recovery	**Sparsity**	**Sub**	**BG**	**Mot**	**Aes**	**Img**	**Dyn**	**AVG5**	**E2E (s)**	**Speedup**
$G_{\text{rec}} = 1$	93.40%	0.9332	0.9533	0.9869	0.5247	0.6809	0.5833	0.8158	**419.344**	**2.46**×
$G_{\text{rec}} = 1$	91.20%	0.9368	0.9525	0.9867	0.5332	0.6838	0.5833	0.8186	435.907	2.37×
$G_{\text{rec}} = 1$	89.17%	0.9319	0.9532	0.9862	0.5414	0.6922	0.5938	0.8210	452.561	2.28×
Per-query, Dense Guard=0	93.40%	0.9360	0.9573	0.9875	0.5441	0.6825	0.5729	0.8215	428.712	2.41×
Per-query, Dense Guard=0	91.20%	0.9342	0.9548	0.9870	0.5440	0.6920	0.5938	0.8224	447.636	2.30×
Per-query, Dense Guard=0	89.17%	0.9340	0.9538	0.9862	0.5522	0.6972	0.5938	**0.8247**	465.129	2.22×
Per-query, Dense Guard=5	93.40%	0.9442	0.9546	0.9883	0.5453	0.6692	0.5625	0.8203	489.139	2.11×
Per-query, Dense Guard=5	91.20%	0.9449	0.9566	0.9884	0.5533	0.6718	0.5313	0.8230	506.179	2.04×
(b) Candidate-Constrained per-query recovery										
Candidate	**Sparsity**	**Sub**	**BG**	**Mot**	**Aes**	**Img**	**Dyn**	**AVG5**	**E2E (s)**	**Speedup**
Draft (0.75), +5	93.40%	0.9157	0.9437	0.9837	0.5172	0.6404	0.5729	0.8001	497.701	2.072×
Draft (0.75), +5	91.20%	0.9033	0.9403	0.9838	0.5240	0.6493	0.5208	0.8001	514.480	2.004×
XAttention, +5	93.40%	0.9401	0.9545	0.9880	0.5441	0.6669	0.5833	0.8187	502.912	2.050×
XAttention, +5	91.20%	0.9428	0.9567	0.9882	0.5561	0.6696	0.5521	0.8227	519.946	1.983×
Draft (0.75)	89.17%	0.8697	0.9251	0.9802	0.5121	0.6496	0.5313	0.7873	474.398	2.173×
XAttention	89.17%	0.9118	0.9460	0.9840	0.5434	0.6850	0.6458	0.8140	480.407	2.146×
Draft (0.5)	89.17%	0.9150	0.9453	0.9856	0.5441	0.6900	0.5521	0.8160	475.866	2.167×
(c) Matched 720p baselines										
Method	**Sparsity**	**Sub**	**BG**	**Mot**	**Aes**	**Img**	**Dyn**	**AVG5**	**E2E (s)**	**Speedup**
FA3 Dense	0.00%	0.9517	0.9592	0.9875	0.5620	0.6738	0.5104	0.8268	1031.087	1.00×
DraftAttention	75.00%	0.8985	0.9366	0.9806	0.5423	0.6462	0.6875	0.8009	664.637	1.55×
XAttention	80.03%	0.8940	0.9358	0.9806	0.5339	0.6342	0.6563	0.7957	562.209	1.83×
Sol-Attn ($\tau = 2$)	88.52%	0.9486	0.9614	0.9884	0.5585	0.6789	0.5313	0.8272	584.440	1.76×
PISA (density $= 0.15$)	85.16%	0.9488	0.9594	0.9880	0.5639	0.6791	0.5104	0.8279	647.646	1.59×
SVG2	89.36%	0.9479	0.9584	0.9870	0.5586	0.6696	0.4688	0.8243	882.280	1.17×

Table 14: **Dense Guard ablation at Wan 720p.** All rows use the same 96-prompt manifest, seed 42, generation configuration, native LightX2V pipeline timing boundary, and six-dimensional VBench evaluation. The Guard=0 videos are byte-identical to the scored no-guard inputs; Guard=5 at $K = 64$ was not part of the scored sweep.

Dense Guard	$K = 39$			$K = 52$			$K = 64$		
	AVG5	**E2E (s)**	**Speedup**	**AVG5**	**E2E (s)**	**Speedup**	**AVG5**	**E2E (s)**	**Speedup**
0	0.8215	428.712	2.41×	0.8224	447.636	2.30×	0.8247	465.129	2.22×
2	0.8187	453.117	2.28×	0.8223	471.249	2.19×	0.8241	487.995	2.11×
5	0.8203	489.139	2.11×	0.8230	506.179	2.04×	0.8230	521.906	1.98×
8	0.8233	524.771	1.96×	0.8249	540.683	1.91×	0.8259	555.471	1.86×
10	0.8231	559.236	1.84×	0.8254	571.407	1.80×	0.8256	581.302	1.77×

F Operator efficiency

F.1 Candidate and recovery operator integration

The candidate generator returns logical KV-block indices and candidate counts for each query block. Full-Domain uses all legal blocks; Draft and XAttention provide smaller candidate sets. The common refiner pools Q/K for its block score, evaluates only the supplied candidates, and selects M_r; the remaining candidate blocks form the recovery set R_r. The backend lowers M_r into its sparse-attention ranges and executes the exact branch once. The interface couples candidate selection to a single exact-attention call.

Recovery reduces K/V to one summary per valid block and combines the contributions from R_r with the exact online-softmax state. For $G_{\text{rec}} = 1$ or 2, a fused epilogue merges the shared recovery states with the backend output. For $G_{\text{rec}} = |G_r|$, the Magi implementation appends compact K/V summaries, lowers each query block to a summary range and its exact ranges, and accumulates both in one query-block kernel launch. The summary mask selects R_r and valid-token counts en-

ter the summary logits. The state construction, merge, and backend requirements are specified in Appendix F.2.

Table 15 reports components measured on the same captured Q/K/V as the full attention call. Scoring and selection includes Q/K pooling and Top-K selection; candidate generation includes the generator's scoring and candidate-index construction. The H200/Magi runs use all 40 captured heads, $d = 128$, Q128/KV128 geometry, $K = 52$, and per-query recovery. XAttention uses $L = 156$ at 480p and $L = 354$ at 720p; Draft uses its variable-size candidate set. The captures contain 32,760 tokens at step 34/layer 14 and 75,600 tokens at step 1/layer 14, respectively. We report CUDA-event medians after five warm-ups and 20 measurements, with each component measured separately on prepared inputs. Scoring and selection takes 3.19–4.03% of the full attention-call time. The full-call measurement includes layout conversion, online selection, recovery, sparse execution, and output restoration.

Table 15: **H200/Magi costs on captured Wan attention.** All times are milliseconds. Share is scoring-and-selection time divided by full attention-call time. Peak extra is the incremental PyTorch allocation during one warmed full call, including output; inputs and runtime buffers are already resident. Mem./dense is this increment divided by that of a matched Magi dense call, including layout conversion and output restoration.

Resolution	Selection	Candidate	Score+select	Share (%)	Full call	Peak extra (MiB)	Mem./dense (%)
480p	Full-Domain	—	0.417	3.87	10.762	1334.1	69.3
480p	Draft Candidate	0.675	0.400	3.42	11.717	1345.4	69.9
480p	XAttention Candidate	1.094	0.381	3.19	11.928	1338.4	69.5
720p	Full-Domain	—	1.121	4.03	27.839	3114.3	70.1
720p	Draft Candidate	2.302	1.201	3.96	30.324	3172.4	71.4
720p	XAttention Candidate	3.800	1.135	3.57	31.841	3129.7	70.5

Costs across models and backends. Table 16 extends the same component measurement to MiniMax-H3/H200 and Wan/B200. H3 uses the native 180-block video partition with tile capacity 256 and an exact 419-token text/audio prefix. Each capture contains four distinct 14-head rank shards; we time them serially and average the four per-rank medians. The full call includes tiling, online selection, backend mapping, recovery, attention, and output restoration. Wan/B200 uses FA4 with Q256/KV128 geometry and the same captured head repeated across 40 heads as Table 25. All rows use per-query recovery, five warm-ups, and 20 measurements. Scoring and selection accounts for 2.13–2.40% of the H3 calls and 5.33–6.61% of the FA4 calls; Table 15 provides the corresponding Full-Domain and Candidate-Constrained breakdowns at both Wan resolutions.

Table 16: **Full-Domain scoring and selection across models and backends.** Times are milliseconds; scoring includes pooling and Top-K. K/B gives selected versus legal video KV blocks. The percentage uses the full call measured on the same inputs in the same run.

Model / GPU / backend	Capture	K/B	Score+select	Full call	Share (%)
MiniMax-H3 / H200 / Magi	Step 0/layer 0	36/180	0.171	7.292	2.34
	Step 0/layer 0	45/180	0.172	8.089	2.13
	Step 1/layer 25	36/180	0.170	7.090	2.40
	Step 1/layer 25	45/180	0.170	7.944	2.15
Wan / B200 / FA4	Step 34/layer 14	39/256	0.292	4.410	6.61
	Step 34/layer 14	52/256	0.315	5.205	6.06
	Step 34/layer 14	64/256	0.323	6.068	5.33

At fixed exact-computation sets, the H200/Magi per-query backend call takes 10.475 ms versus 9.849 ms for $G_{\mathrm{rec}} = 1$ recovery (Table 24). Generation-level E2E includes the complete online attention computation and is reported in the main results.

Memory scaling. For H heads, N tokens, head dimension d, and $B = \lceil N/B_{\mathrm{KV}} \rceil$ KV blocks, recovery stores $O(HBd)$ key/value summaries and $O(B)$ valid-token counts. Per-query recovery uses each query directly with these summaries; the fused implementation accumulates their contributions without storing an $N \times N$ attention matrix. Candidate metadata scales as $O(HB_qL)$ for B_q query blocks and candidate capacity L, while the Full-Domain block scorer uses $O(HB_qB)$ scores.

The fused Magi implementation combines token permutation with packing, shares the resulting Q/K storage with selection, and reuses Q storage for output restoration. Candidate-generator workspaces and backend descriptors contribute to the full attention-call peak.

F.2 Recovery operators across query granularities

Shared formulation. The $G_{\mathrm{rec}} = 1$, $G_{\mathrm{rec}} = 2$, and $G_{\mathrm{rec}} = |G_r|$ configurations share the candidate KV-block set, the selected exact-computation blocks, and the selected-block count, and differ in the query granularity used to approximate omitted blocks. For a query block G_r, use the selection domain D_r, $M_r \subseteq D_r$ its exact-computation set, and $R_r = D_r \setminus M_r$ its recovery set. For the valid tokens in block B_b, we store one key/value summary per head,

$$n_b = |B_b|, \qquad \bar{k}_b = \frac{1}{n_b} \sum_{j \in B_b} k_j, \qquad \bar{v}_b = \frac{1}{n_b} \sum_{j \in B_b} v_j. \tag{12}$$

For a partial tail block, n_b denotes its valid-token count. Given a partition $G_r = \dot{\bigcup}_g G_{r,g}$, each group forms

$$\bar{q}_{r,g} = \frac{1}{|G_{r,g}|} \sum_{i \in G_{r,g}} q_i, \qquad \rho_{r,g,b} = \frac{\bar{q}_{r,g}^{\top} \bar{k}_b}{\sqrt{d}}, \tag{13}$$

and represents every $b \in R_r$ by a summary logit $\rho_{r,g,b} + \log n_b$ and value vector $\bar{v}_b$. Equivalently, its stable online-softmax state is

$$\begin{aligned} m_{r,g}^R &= \max_{b \in R_r} \rho_{r,g,b}, \\ \ell_{r,g}^R &= \sum_{b \in R_r} n_b \exp(\rho_{r,g,b} - m_{r,g}^R), \\ c_{r,g}^R &= \sum_{b \in R_r} n_b \exp(\rho_{r,g,b} - m_{r,g}^R) \bar{v}_b. \end{aligned} \tag{14}$$

If exact attention supplies (m_i^M, ℓ_i^M, c_i^M), the two states are combined in FP32 with $m_i = \max(m_i^M, m_{r,g(i)}^R)$,

$$\begin{aligned} \ell_i &= e^{m_i^M - m_i} \ell_i^M + e^{m_{r,g(i)}^R - m_i} \ell_{r,g(i)}^R, \\ c_i &= e^{m_i^M - m_i} c_i^M + e^{m_{r,g(i)}^R - m_i} c_{r,g(i)}^R, \qquad \widehat{O}_i = c_i / \ell_i. \end{aligned} \tag{15}$$

All granularities therefore share D_r, M_r, and the normalization rule.

Shared recovery states. With $G_{\mathrm{rec}} = 1$, one query-block mean and one recovery state are reused by every valid query in the Q128 block. With $G_{\mathrm{rec}} = 2$, the same parent-block domain and exact-computation set are retained, but recovery scoring is partitioned into up to two fixed-width Q64 groups; the final group uses the actual valid query count. In the evaluated modular Magi implementation, the block summaries first build one or two side states, respectively; Magi executes exact attention on M_r, and a fused epilogue performs Eq. (15) and the output cast. The $G_{\mathrm{rec}} = 2$ configuration selects the state associated with each query's Q64 group, while both groups inherit D_r and M_r from their parent query block.

Per-query recovery. Per-query recovery uses $G_{\mathrm{rec}} = |G_r|$ and $G_{r,g(i)} = \{i\}$. Hence

$$\mu_{i,b} = \frac{q_i^{\top} \bar{k}_b}{\sqrt{d}}, \qquad \widetilde{Z}_i^R = \sum_{b \in R_r} n_b e^{\mu_{i,b}}, \qquad \widetilde{N}_i^R = \sum_{b \in R_r} n_b e^{\mu_{i,b}} \bar{v}_b. \tag{16}$$

All per-query states are evaluated within a single query-block launch. The Key/value summaries are appended once per head as a compact suffix, and each query block is lowered to one summary range followed by its exact token ranges. The summary mask identifies R_r; GMMA evaluates the query-block-by-summary matrix, adds $\log n_b$ to valid summary logits, and updates the same online-softmax accumulator as the exact ranges. Assigning the tile to one owner CTA fuses the summary evaluation and exact-state composition in a single kernel.

Table 17: **Recovery operator realizations.** The exact branch and block summaries are common, while the number of queries represented by each recovery state and the summary computation vary. The logical per-query states reside in the tiled online-softmax accumulator.

Grouping	G_{rec}	Query statistic	Exact-computation set	Recovery execution
One query group	1	Q128 mean $\bar{q}_r$	shared M_r	one recovery state per query block; fused state merge/cast
Two query groups	2	Q64 means $\bar{q}_{r,g}$	shared M_r	up to two recovery states per query block; group-mapped fused merge/cast
One group per query	$\lvert G_r\rvert$	token query q_i	shared M_r	one batched summary range in the existing tiled kernel; same online-softmax accumulator

This mapping preserves the identity layout of the original K/V tokens. Fixed-cardinality selected sets retain the order emitted by the selector, whereas variable-cardinality selected sets are reordered into a common execution layout. Consecutive descriptors with the same Q range are coalesced by run-length grouping. These operations modify the execution descriptors while preserving the selected exact-computation blocks. Thus $G_{\mathrm{rec}} = |G_r|$ denotes query-specific recovery weights, while the execution granularity remains the query block.

Approximation relation. For token logit $s_{i,j} = q_i^\top k_j/\sqrt{d}$, define the group center $\nu_{r,b} = \bar{q}_r^\top \bar{k}_b/\sqrt{d}$. Since $\mu_{i,b} = n_b^{-1}\sum_{j\in B_b} s_{i,j}$, the following decomposition is exact:

$$\frac{1}{n_b}\sum_{j\in B_b}(s_{i,j}-\nu_{r,b})^2 = \frac{1}{n_b}\sum_{j\in B_b}(s_{i,j}-\mu_{i,b})^2 + (\mu_{i,b}-\nu_{r,b})^2, \tag{17}$$

where $\mu_{i,b} - \nu_{r,b} = (q_i - \bar{q}_r)^\top \bar{k}_b/\sqrt{d}$. For the nested partition, refining $G_{\mathrm{rec}} = 1$ to $G_{\mathrm{rec}} = 2$ reduces the residual caused by sharing a query mean in aggregate, and per-query recovery removes it. Equation (17) therefore establishes a monotonic refinement of the block-logit approximation. Final attention error and video quality additionally depend on value dispersion, cross-block softmax normalization, and error cancellation.

Bounds on the normalization term. For a valid block, the exact block denominator is

$$Z_{i,b} = \sum_{j\in B_b} e^{s_{i,j}}, \qquad \widetilde{Z}_{i,b} = n_b e^{\mu_{i,b}}. \tag{18}$$

Because $\mu_{i,b} = n_b^{-1}\sum_{j\in B_b} s_{i,j}$, Jensen's inequality gives $\widetilde{Z}_{i,b} \le Z_{i,b}$. If $\Delta_{i,b} = \max_{j\in B_b}|s_{i,j} - \mu_{i,b}|$, then

$$\widetilde{Z}_{i,b} \le Z_{i,b} \le e^{\Delta_{i,b}}\widetilde{Z}_{i,b}. \tag{19}$$

Thus per-query recovery evaluates the approximation at the query-specific mean logit, giving a lower bound on the exact normalization term by Jensen's inequality. The $G_{\mathrm{rec}} = 1$ case shifts this mean by $\mu_{i,b} - \nu_{r,b}$.

Complexity and storage. For T_q queries, B legal blocks, selected-block count K, and head dimension d, the recovery-score work is respectively $O((B-K)d)$, $O(2(B-K)d)$, and $O(T_q(B-K)d)$ for $G_{\mathrm{rec}} = 1$, $G_{\mathrm{rec}} = 2$, and $G_{\mathrm{rec}} = |G_r|$, respectively. The per-query computation is realized as one tiled matrix product within the query-block launch. All queries share $O(Bd)$ key/value-summary storage per head. Relative to token-level exact attention over the omitted region, the summary key axis shrinks from approximately $(B-K)T_k$ tokens to $B-K$ entries. The $G_{\mathrm{rec}} = |G_r|$ configuration therefore trades additional batched summary-score arithmetic for a finer block-logit approximation.

Backend requirements and candidate domains. The mathematical state in Eqs. (14)– (15) is backend-independent and can wrap an exact executor when $G_{\mathrm{rec}} \in \{1, 2\}$. The fused $G_{\mathrm{rec}} = |G_r|$ path additionally requires a backend to admit summary ranges and masks into its tiled online-softmax update. For Full-Domain, D_r is the complete legal block domain. For Candidate-Constrained execution, the same equations use $D_r = S_r$ and recover only $S_r \setminus M_r$; approximation outside S_r remains governed by candidate-set coverage. The evaluated per-query configuration

Table 18: **Operator analysis at fixed selected blocks across execution backends.**

Backend	Selected set	Sparsity	Kernel (ms)	Kernel speedup	Complete call (ms)	Call speedup	Call vs. FA3	Rel-L2 (FA3)
FlexAttention	Dense	0.00%	57.896	1.000×	57.738	1.000×	–	0.00109
FlexAttention	Full-Domain	79.69%	11.654	4.968×	12.651	4.564×	2.272×	0.22306
FlexAttention	XAttention Candidate	79.69%	11.763	4.922×	12.919	4.469×	2.225×	0.22306
BSA	Full-Domain	79.69%	13.548	4.273×	14.611	3.952×	1.956×	0.22307
BSA	XAttention Candidate	79.69%	13.734	4.216×	15.072	3.831×	1.896×	0.22307
Magi	Full-Domain	79.69%	10.176	**5.689**×	11.949	**4.832**×	**2.537**×	0.22306
Magi	XAttention Candidate	79.69%	10.262	**5.642**×	12.032	**4.799**×	**2.519**×	0.22306
FlexAttention	Draft Candidate	79.69%	11.614	4.985×	12.632	4.571×	2.246×	0.22311

uses a shared M_r with query-specific recovery weights. Row-dependent candidate KV-block sets or exact-computation sets can be represented by row-wise masks within the same query-block launch, defining a distinct extension of the operator.

In Table 18, latencies are CUDA-event medians after 5 warm-up iterations and 20 rotating measurements; loading, host-to-device transfer, and compilation are not included. Kernel and call speedups use the common Flex Dense row; "Call vs. FA3" uses the matched FA3 Dense timing from each execution batch. "Rel-L2 (FA3)" measures output error relative to FA3 Dense. The fixed-set call starts after exact-computation-block construction, as described in Appendix C. On the same exact-computation set, FlexAttention and Magi agree with their reference execution within 4.47×10^{-4}–4.67×10^{-4} relative L2, whereas Rel-L2 (FA3) measures the dense-set approximation error.

F.3 SOL-ATTN AND MODULAR RECOVERY EVIDENCE

Table 19: **Sol-Attn quality and latency.** Each entry pairs scores and E2E from the same generation batches. All four thresholds use 96 prompts.

τ	N	AVG5	Dynamic	E2E (s)
0.5	96	0.836786	0.343750	230.931
1	96	0.836001	0.333333	217.142
2	96	0.834316	0.354167	200.498
2.5	96	0.826244	0.333333	195.399

Table 20: **E2E on 42 matched seed-42 prompts.** QUASAR uses Full-Domain selection.

Method	E2E (s)
Sol-Attn $\tau = 0.5$	242.096
Sol-Attn $\tau = 1$	227.524
Sol-Attn $\tau = 2$	209.476
Sol-Attn $\tau = 2.5$	204.127
QUASAR ($G_{\text{rec}} = 1$, 84.77% sparse)	177.460
QUASAR ($G_{\text{rec}} = 2$, 84.77% sparse)	176.519
QUASAR ($G_{\text{rec}} = 1$, 79.69% sparse)	182.112
QUASAR ($G_{\text{rec}} = 2$, 79.69% sparse)	183.773
QUASAR ($G_{\text{rec}} = 1$, 75.00% sparse)	191.963
QUASAR ($G_{\text{rec}} = 2$, 75.00% sparse)	190.701

Reconstruction accuracy at different recovery granularities. Table 21 holds captured Q/K/V tensors and the exact-computation set fixed while varying the number of queries represented by each recovery state. Wan uses all 40 heads and 32,760 valid queries; both H3 tensors use all four ranks, 56 heads, and 37,715 queries, retaining the 419-token prefix. The controlled $G_{\text{rec}} = 2$ configuration divides each logical query block evenly, including a 60+60 split of the Wan tail; the generation implementation uses fixed-width Q64 groups and a 64+56 tail split.

Table 21: **Recovery granularity and attention-output error.** Relative Frobenius error to the matched Magi Dense output. H3 sparsity is measured over 180 video blocks; the 419-token prefix remains exact.

Captured tensors	Sparsity (%)	Exact-only	$G_{\mathrm{rec}} = 1$	$G_{\mathrm{rec}} = 2$	$G_{\mathrm{rec}} = 4$	$G_{\mathrm{rec}} = \lvert G_r \rvert$
Wan	84.77	0.244772	0.225546	0.225052	0.224570	0.221035
Wan	79.69	0.203610	0.188967	0.188588	0.188212	0.185300
Wan	75.00	0.175278	0.163561	0.163246	0.162934	0.160435
H3 step 0/layer 0	85.00	0.195251	0.188494	0.188220	0.187014	0.185484
H3 step 0/layer 0	80.00	0.151185	0.150969	0.150668	0.149332	0.147318
H3 step 0/layer 0	75.00	0.122497	0.125683	0.125621	0.125138	0.122773
H3 step 1/layer 25	85.00	0.216530	0.227330	0.226449	0.224451	0.207742
H3 step 1/layer 25	80.00	0.187319	0.200627	0.199708	0.197615	0.181331
H3 step 1/layer 25	75.00	0.168374	0.183130	0.182203	0.180064	0.163914

Per-query recovery on H3. For the H3 step-1/layer-25 capture at 80% video-block sparsity, per-query recovery reduces Rel-L2 from 0.200627 with $G_{\mathrm{rec}} = 1$ recovery to 0.181331, using the same exact-computation set and KV summaries. The exact-only result is 0.187319.

Refining KV summaries at fixed exact computation. Query grouping controls sharing across queries, while KV block size controls how finely keys and values are summarized. Table 22 subdivides KV128 blocks into KV64 and KV32 summaries while preserving every exactly computed token. We sample heads 0, 5, 19, and 39 and eight evenly spaced complete query blocks, with eight evenly spaced queries per block, from each Wan capture. The reference calculation uses FP32 and $K = 52$ original KV128 blocks. Per-query error decreases at each refinement in both captures, alongside lower within-block logit variation. The $G_{\mathrm{rec}} = 1$ and per-query rows show how KV refinement interacts with query sharing.

Table 22: **KV-summary refinement on captured Wan attention.** Rel-L2 to exact dense attention on 256 sampled head/query pairs. Exact token sets are fixed across rows; the number of summaries increases with refinement.

Capture	Recovery	KV128	KV64	KV32
480p, step 34/layer 14	$G_{\mathrm{rec}} = 1$	0.162607	0.162453	0.162238
480p, step 34/layer 14	Per-query	0.157367	0.155045	0.152907
720p, step 1/layer 14	$G_{\mathrm{rec}} = 1$	0.222351	0.223748	0.224492
720p, step 1/layer 14	Per-query	0.208438	0.205411	0.203051

Modular composition and execution cost. Magi and FlexAttention combine the same recovery states with their exact outputs on the captured Wan tensors. Across 40 heads, three sparsity levels, and five output variants, the maximum aggregate backend relative-Frobenius difference is 0.039815%. The recovery state can therefore be reused across exact executors.

Table 23: **Recovery latency at fixed selected blocks on H200.** CUDA-event medians, three warm-ups and 20 rotating measurements.

Sparsity (%)	Exact-only (ms)	$G_{\mathrm{rec}} = 1$ (ms)	$G_{\mathrm{rec}} = 2$ (ms)	$G_{\mathrm{rec}} = 4$ (ms)
84.77	7.928480	8.050768	8.108096	8.154176
79.69	9.550848	9.643008	9.669248	9.758224
75.00	11.065664	11.201328	11.316608	11.486368

Table 23 uses one captured attention head replicated across 40 heads and includes layout conversion, value-block reduction, recovery-state construction, Magi execution, merging, and output restoration. Relative to $G_{\mathrm{rec}} = 1$, $G_{\mathrm{rec}} = 2$ adds 0.026–0.115 ms and $G_{\mathrm{rec}} = 4$ adds 0.103–0.285 ms. Table 24 gives the per-query comparison at 79.69% sparsity.

Table 24: **Per-query Magi/H200 tiled-summary timing at 79.69% sparsity.** CUDA-event medians, three warm-ups and 30 repeats on the same exact-computation set.

Implementation	Backend (ms)	Δ vs $G_{\mathrm{rec}} = 1$ (ms)
$G_{\mathrm{rec}} = 1$ Magi	9.849056	-
$G_{\mathrm{rec}} = \lvert G_r \rvert$ tiled-summary backend	10.475376	$+0.626$

Generation-level E2E for the same $G_{\mathrm{rec}} = |G_r|$ configuration is reported in Table 5.

End-to-end attention-call timing. An additional benchmark starts from original-order contiguous BF16 Q/K/V and includes the full attention call, from Q/K pooling, scoring, and selection through layout conversion, exact execution, value reduction, recovery, and output restoration. The $G_{\mathrm{rec}} = 1$ Magi computation takes 9.282, 11.440, and 12.966 ms at 84.77%, 79.69%, and 75.00% sparsity. On the same captured tensors, Sol-Attn diagonal-threshold calls take 15.313, 12.202, and 7.766 ms at $\tau = 0.5, 1, 2$, with relative output errors 0.111664, 0.140965, and 0.212474.

Per-query recovery on B200/FA4. Table 25 evaluates the full attention call with Q256/KV128 geometry on BF16 Wan inputs of shape $[32760, 40, 128]$. One captured head is repeated across 40 heads. Each paired run fixes the exact-computation set and alternates exact-only and per-query recovery for five warm-ups and 20 measurements. The full call includes layout conversion, scoring and selection, backend mapping, summary packing, attention, and output restoration. Per-query recovery improves output accuracy across all three budgets with 0.585–0.668 ms additional latency.

Table 25: **B200/FA4 recovery on captured Wan Q/K/V.** CUDA-event medians; Rel-L2 uses the same-input FA4 Dense reference.

K	Exact-only (ms)	Per-query (ms)	Exact-only Rel-L2	Per-query Rel-L2
39	3.609632	4.194176	0.274792	0.268425
52	4.453952	5.087616	0.232193	0.227893
64	5.192336	5.860064	0.196294	0.193472

F.4 Additional generation results

Table 26 provides an independent MiniMax-H3 replication. Table 27 reports 21 Wan configurations on 96 seed-42 prompts, complementing the per-query results in Table 9.

Table 26: **MiniMax-H3 seed-42 replication on 96 prompts.** All three selection configurations use BSA and Q128/KV128.

Method	Sparsity	AVG5	E2E (s)	Speedup
Dense	0.00%	0.835780	133.794	1.000×
QUASAR-Full	79.79%	0.829590	99.291	1.348×
QUASAR-XAttention	79.79%	0.830013	106.065	1.261×

Table 27: **Additional Wan 480p sparsity sweep.** Recovery rows use $G_{\mathrm{rec}} = 1$.

Selection	Sparsity (%)	Recovery	Sub	BG	Mot	Aes	Img	Dyn	AVG5	E2E (s)
Full-Domain	89.84	Exact-only	0.9214	0.9436	0.9874	0.5153	0.5955	0.5104	0.792640	167.832
Full-Domain	89.84	$G_{\mathrm{rec}} = 1$	0.9576	0.9602	0.9908	0.5295	0.6850	0.3438	0.824618	167.436
Full-Domain	84.77	Exact-only	0.9347	0.9520	0.9888	0.5319	0.6217	0.5104	0.805784	174.524
Full-Domain	84.77	$G_{\mathrm{rec}} = 1$	0.9604	0.9618	0.9911	0.5406	0.6906	0.4062	0.828898	177.262
Full-Domain	79.69	Exact-only	0.9431	0.9552	0.9898	0.5441	0.6281	0.4792	0.812068	182.623
Full-Domain	79.69	$G_{\mathrm{rec}} = 1$	0.9658	0.9652	0.9914	0.5440	0.6860	0.3854	0.830496	182.075
Full-Domain	75.00	Exact-only	0.9520	0.9596	0.9901	0.5516	0.6347	0.4479	0.817627	188.592
Full-Domain	75.00	$G_{\mathrm{rec}} = 1$	0.9690	0.9661	0.9917	0.5527	0.6823	0.3958	0.832352	191.722
Full-Domain	69.92	Exact-only	0.9582	0.9619	0.9908	0.5513	0.6431	0.4792	0.821071	196.961
Full-Domain	69.92	$G_{\mathrm{rec}} = 1$	0.9700	0.9683	0.9918	0.5545	0.6818	0.3854	0.833283	196.820
Full-Domain	64.84	$G_{\mathrm{rec}} = 1$	0.9692	0.9693	0.9917	0.5561	0.6772	0.3958	0.832682	204.103
Full-Domain	59.77	Exact-only	0.9653	0.9653	0.9914	0.5596	0.6438	0.4062	0.825080	211.548
Full-Domain	59.77	$G_{\mathrm{rec}} = 1$	0.9702	0.9685	0.9916	0.5597	0.6750	0.3958	0.833008	211.500
Draft	84.77	Exact-only	0.9300	0.9497	0.9877	0.5291	0.6138	0.5312	0.802076	179.061
Draft	84.77	$G_{\mathrm{rec}} = 1$	0.9492	0.9565	0.9896	0.5404	0.6707	0.4375	0.821275	179.378
Draft	79.69	Exact-only	0.9375	0.9522	0.9884	0.5369	0.6301	0.5000	0.809005	185.270
Draft	79.69	$G_{\mathrm{rec}} = 1$	0.9539	0.9597	0.9899	0.5463	0.6739	0.4688	0.824728	188.861
Draft	75.00	Exact-only	0.9451	0.9570	0.9893	0.5453	0.6326	0.5104	0.813865	192.272
Draft	75.00	$G_{\mathrm{rec}} = 1$	0.9576	0.9621	0.9905	0.5464	0.6677	0.4167	0.824856	192.679
XAttention (60.94%)	79.69	Exact-only	0.9443	0.9555	0.9897	0.5432	0.6311	0.4792	0.812744	200.497
XAttention (60.94%)	79.69	$G_{\mathrm{rec}} = 1$	0.9595	0.9625	0.9908	0.5486	0.6636	0.4375	0.824995	201.098

G CANDIDATE SELECTION WITH EXACT-ONLY ATTENTION

Table 28 reports the matched seed-42 exact-only results. Draft uses its own always-included blocks and variable-size candidate rule, so its candidate fraction is averaged over query blocks.

Table 28: **Candidate coverage with exact-only attention.**

Candidate	**Candidate analysis**			**Generation**	
	Candidate frac.	**Sparsity**	**Selection recall↑**	**AVG5↑**	**E2E (s)↓**
Draft	53.38%	84.77%	99.932%	0.8021	179.061
Draft	53.38%	79.69%	99.786%	0.8090	185.270
Draft	53.38%	75.00%	99.109%	0.8139	192.272
XAttention	30.47%	79.69%	99.992%	0.8116	200.182
XAttention	40.63%	79.69%	100.000%	0.8124	200.247
XAttention	60.94%	79.69%	100.000%	0.8127	200.497
Random	30.47%	79.69%	30.389%	0.7281	183.509
Random	40.63%	79.69%	40.554%	0.7738	183.424
Random	60.94%	79.69%	60.956%	0.7941	183.561

H CANDIDATE COVERAGE WITH ALWAYS-INCLUDED BLOCKS

Fix a row r and a reference budget $1 \le K \le N = |\Omega_\Gamma|$, with $F_r = M_r^{\mathrm{FD}}$ the Full-Domain Top-K and $a^\Gamma_{r,(j)}$ the jth largest true score. For always-included candidate blocks $H_r \subseteq \Omega_\Gamma$, let

$$U_r = \Omega_\Gamma \setminus H_r, \qquad S_r^{\mathrm{CC}} = H_r \cup \mathrm{TopK}^{\ell_r}_{b \in U_r}(\widehat{a}_{r,b}), \qquad T_r = F_r \setminus H_r, \quad k_r^{\mathrm{res}} = |T_r|. \tag{20}$$

The set T_r is the true-score Top-k_r^{res} of U_r. Apply Theorem 1 and the partial-coverage result below with $(K, L, F_r, \Omega_\Gamma) \mapsto (k_r^{\mathrm{res}}, \ell_r, T_r, U_r)$, using ranks and relative errors within U_r. Structural membership guarantees candidate inclusion; the common final Top-K determines exact computation. Any residual missed-block bound m gives $\mathrm{Recall}_r\,@K \ge 1 - m/K$ for the full target set. Draft's first-frame and same-frame blocks form H_r; its remaining ranking supplies the residual candidates.

H.1 PROOF OF CANDIDATE COVERAGE

Proof. Every $b \in F_r$ has $\widehat{a}_{r,b} \ge a^\Gamma_{r,(K)} + \min_{u \in F_r} e_{r,u}$. Among competitors $c \notin F_r$, at most $L - K - t$ have $a^\Gamma_{r,c} > a^\Gamma_{r,(L-t+1)}$, and at most t have $\delta_{r,c} > \delta_{r,[t+1]}$. Every remaining competitor satisfies

$$\widehat{a}_{r,c} \le a^\Gamma_{r,(L-t+1)} + \min_{u \in F_r} e_{r,u} + \delta_{r,[t+1]} < a^\Gamma_{r,(K)} + \min_{u \in F_r} e_{r,u} \le \widehat{a}_{r,b}. \tag{21}$$

Thus at most $L-K$ competitors can precede any target b in the proxy ordering. Including the other $K-1$ targets, at most $L-1$ blocks precede b, so every target lies in the proxy Top-L and hence in S_r^{CC}. Under the common final score and tie rule, any included target precedes every non-target. Therefore $F_r \cap M_r^{\mathrm{CC}} = F_r \cap S_r^{\mathrm{CC}}$, giving $M_r^{\mathrm{CC}} = F_r$ under complete coverage. □

H.2 Partial coverage

For $0 \le m < K$, let $F_r^{(m)}$ contain the true-score top $K-m$ blocks. Define

$$\delta_{r,c}^{(m)} = e_{r,c} - \min_{b \in F_r^{(m)}} e_{r,b}, \qquad c \notin F_r^{(m)}, \tag{22}$$

with $\delta_{r,[j]}^{(m)}$ the jth largest of these values. For $K \le L$ and $L+m < N$, if some $t \in \{0, \dots, L-K+m\}$ satisfies

$$a_{r,(K-m)}^{\Gamma} - a_{r,(L+m-t+1)}^{\Gamma} > \delta_{r,[t+1]}^{(m)}, \tag{23}$$

then $|F_r \setminus M_r^{\mathrm{CC}}| \le m$ and $\mathrm{Recall}_r\,@K \ge 1 - m/K$.

Proof. Suppose more than m targets are missed. The retention identity above implies that more than m targets are absent from S_r^{CC}, hence from its proxy Top-L subset. At least one missing target b belongs to $F_r^{(m)}$, and the proxy Top-L contains at least $L-K+m+1$ blocks outside F_r, all preceding b. Among blocks outside F_r, at most $L-K+m-t$ have true score above $a_{r,(L+m-t+1)}^{\Gamma}$. At most t have relative error above $\delta_{r,[t+1]}^{(m)}$, since they are a subset of $\Omega_\Gamma \setminus F_r^{(m)}$. Every other such block c satisfies

$$\widehat{a}_{r,c} \le a_{r,(L+m-t+1)}^{\Gamma} + \min_{u \in F_r^{(m)}} e_{r,u} + \delta_{r,[t+1]}^{(m)} < a_{r,(K-m)}^{\Gamma} + \min_{u \in F_r^{(m)}} e_{r,u} \le \widehat{a}_{r,b}. \tag{24}$$

Only $L-K+m$ blocks outside F_r can precede b, a contradiction. □

Uniform-error specialization. If $|e_{r,b}| \le \epsilon_r$ for all blocks, then every relative error is at most $2\epsilon_r$. Taking $t=0$ recovers the complete-coverage condition $a_{r,(K)}^{\Gamma} - a_{r,(L+1)}^{\Gamma} > 2\epsilon_r$, and gives the partial-coverage condition

$$a_{r,(K-m)}^{\Gamma} - a_{r,(L+m+1)}^{\Gamma} > 2\epsilon_r. \tag{25}$$

For the same valid ϵ_r, this partial condition uses the allowed number of missed targets to enlarge the comparison rank from $L+1$ to $L+m+1$. A common additive shift of the proxy scores leaves all relative errors unchanged. The factor 2 is sharp when only a uniform error radius is known: two blocks with errors $-\epsilon_r$ and $+\epsilon_r$ can reverse any positive true-score gap smaller than $2\epsilon_r$.

The uniform-error diagnostics in Figure 5 and Table 8 use the residual radius $\epsilon_r^U = \max_{b \in U_r} |e_{r,b}|$ and the per-block set

$$\mathcal{A}_r^U = \left\{ b \in T_r : a_{r,b}^{\Gamma} - a_{r,(\ell_r+1;U_r)}^{\Gamma} \le 2\epsilon_r^U \right\}, \qquad F_r \setminus S_r^{\mathrm{CC}} \subseteq \mathcal{A}_r^U, \qquad \mathrm{Recall}_r\,@K \ge 1 - \frac{|\mathcal{A}_r^U|}{K}. \tag{26}$$

Here $(j; U_r)$ denotes rank j within U_r. Any $b \in T_r \setminus \mathcal{A}_r^U$ outranks every block outside the true-score top ℓ_r in the proxy ordering, proving the inclusion. For fixed L, set $H_r = \varnothing$ and $\ell_r = L$. With the same valid uniform radius, the partial condition above retains every guarantee from this per-block bound: if at least $K-m$ targets exceed $a_{r,(L+1)}^{\Gamma} + 2\epsilon_r$, then $a_{r,(K-m)}^{\Gamma} - a_{r,(L+m+1)}^{\Gamma} > 2\epsilon_r$ whenever $L+m < N$; otherwise coverage follows from the cardinality bound below.

Boundary cases and ties. When $L=N$, coverage is complete; when $L=K$, Theorem 1 uses $t=0$. If $L+m \ge N$, then $|F_r \setminus S_r^{\mathrm{CC}}| \le N-L \le m$ directly, so no score at rank beyond N is required. The case $m=K$ gives the immediate bound $\mathrm{Recall}_r\,@K \ge 0$. For structural candidates, $k_r^{\mathrm{res}} = 0$ or $\ell_r = |U_r|$ gives complete coverage, with $\mathcal{A}_r^U = \varnothing$. When $\ell_r < k_r^{\mathrm{res}}$, one may apply the residual result to the top ℓ_r targets and count the remaining $k_r^{\mathrm{res}} - \ell_r$ targets in the missed-block bound; $\ell_r = 0$ retains all targets in H_r. If $|S_r^{\mathrm{CC}}| < K$, the final budget is $K_r = \min(K, |S_r^{\mathrm{CC}}|)$ and all available candidates are selected, preserving the retention identity. All orderings use descending score and ascending logical block ID. The strict inequalities make the score comparisons in both proofs independent of ties; at equality, the fixed tie rule applies.

Offline score analysis. For each captured row, the analysis pairs Full-Domain block scores with candidate scores under the frozen positive, order-preserving calibration described in Appendix D. These paired scores supply the relative-error order statistics for evaluating complete-coverage conditions and partial-recall bounds offline. Online candidate generation and final selection follow Algorithm 1.

I OUTPUT ERROR FROM CHANGES IN THE SELECTED BLOCKS

More generally, let O_i^{Ω} be exact attention over the legal Full-Domain, and let O_i^{FD} and O_i^{CC} be exact attention restricted to the corresponding QUASAR exact-computation sets. The total output difference has the exact decomposition

$$O_i^{\Omega} - O_i^{\mathrm{CC}} = \big(O_i^{\Omega} - O_i^{\mathrm{FD}}\big) + \big(O_i^{\mathrm{FD}} - O_i^{\mathrm{CC}}\big), \tag{27}$$

and hence the norm bound

$$\big\|O_i^{\Omega} - O_i^{\mathrm{CC}}\big\| \le \underbrace{\big\|O_i^{\Omega} - O_i^{\mathrm{FD}}\big\|}_{\text{Full-Domain sparsification error}} + \underbrace{\big\|O_i^{\mathrm{FD}} - O_i^{\mathrm{CC}}\big\|}_{\text{output error caused by candidate-induced selected-set changes}}. \tag{28}$$

The Full-Domain sparsification term and the approximation from value-block summaries are analyzed separately.

For a reference selected set Y_r, the disjoint decomposition is

$$Y_r \setminus M_r^{\mathrm{CC}} = (Y_r \setminus S_r^{\mathrm{CC}}) \,\dot\cup\, ((Y_r \cap S_r^{\mathrm{CC}}) \setminus M_r^{\mathrm{CC}}). \tag{29}$$

From Eq. (4),

$$\big\|O_i^{\mathrm{FD}} - O_i^{\mathrm{CC}}\big\|_2 = \eta_i^{\mathrm{mig}} \big\|\mu_i^{+} - \mu_i^{-}\big\|_2 . \tag{30}$$

Fix $i \in G_r$, and abbreviate the Full-Domain and Candidate-Constrained exact-computation sets as $F = M_{r,K}^{\mathrm{FD}}$ and $C = M_{r,K}^{\mathrm{CC}}$. This section compares exact attention restricted to these selected sets; recovery is analyzed separately in Section K. For any legal block, define its exact unnormalized attention mass and the mass of a block set by

$$w_{i,b} = \sum_{j \in B_b \cap \mathcal{V}_r} e^{q_i^\top k_j/\sqrt{d}}, \qquad W_i(A) = \sum_{b \in A} w_{i,b}. \tag{31}$$

Assume $W_i(F), W_i(C) > 0$, and define the block distributions induced by F and C

$$p_{i,b}^{F} = \frac{\mathbf{1}[b \in F] w_{i,b}}{W_i(F)}, \qquad p_{i,b}^{C} = \frac{\mathbf{1}[b \in C] w_{i,b}}{W_i(C)}. \tag{32}$$

Theorem 3 (Output error from changes in the selected blocks)**.** *Let* $\eta_i^{\mathrm{mig}} = \mathrm{TV}(p_i^F, p_i^C)$. *Then*

$$\eta_i^{\mathrm{mig}} = \max\left\{ \frac{W_i(F \setminus C)}{W_i(F)}, \frac{W_i(C \setminus F)}{W_i(C)} \right\}. \tag{33}$$

For $w_{i,b} > 0$*, let*

$$v_{i,b}^{c} = \frac{\sum_{j \in B_b \cap \mathcal{V}_r} e^{q_i^\top k_j/\sqrt{d}} v_j}{w_{i,b}}. \tag{34}$$

If $\eta_i^{\mathrm{mig}} > 0$*, define*

$$\mu_i^{+} = \frac{1}{\eta_i^{\mathrm{mig}}} \sum_b \big(p_{i,b}^F - p_{i,b}^C\big)_+ v_{i,b}^c, \qquad \mu_i^{-} = \frac{1}{\eta_i^{\mathrm{mig}}} \sum_b \big(p_{i,b}^C - p_{i,b}^F\big)_+ v_{i,b}^c. \tag{35}$$

The exact outputs restricted to F *and* C *obey*

$$O_i^{\mathrm{FD}} - O_i^{\mathrm{CC}} = \eta_i^{\mathrm{mig}} (\mu_i^{+} - \mu_i^{-}), \tag{36}$$

and therefore admit the exact error characterization

$$\big\|O_i^{\mathrm{FD}} - O_i^{\mathrm{CC}}\big\|_2 = \eta_i^{\mathrm{mig}} \big\|\mu_i^{+} - \mu_i^{-}\big\|_2 . \tag{37}$$

A generic value-centroid relaxation is

$$\left\|O_i^{\mathrm{FD}} - O_i^{\mathrm{CC}}\right\|_2 \le \eta_i^{\mathrm{mig}} \max_{\substack{b,c\in F\cup C \\ w_{i,b}w_{i,c}>0}} \left\|v_{i,b}^c - v_{i,c}^c\right\|_2 . \tag{38}$$

When $\eta_i^{\mathrm{mig}} = 0$, the two outputs are equal.

Proof. On $F\cap C$, both distributions are proportional to the same positive block masses. If $W_i(F) \ge W_i(C)$, then $p_{i,b}^F \le p_{i,b}^C$ on the intersection; the inequality reverses when $W_i(F) \le W_i(C)$. Expanding $\frac{1}{2}\sum_b |p_{i,b}^F - p_{i,b}^C|$ over $F\setminus C$, $C\setminus F$, and $F\cap C$, and substituting $W_i(F) = W_i(F\cap C) + W_i(F\setminus C)$ and its counterpart for C, yields Eq. (33). Moreover,

$$O_i^{\mathrm{FD}} - O_i^{\mathrm{CC}} = \sum_b (p_{i,b}^F - p_{i,b}^C) v_{i,b}^c. \tag{39}$$

The positive and negative parts of $p_i^F - p_i^C$ each sum to η_i^{mig}, giving the exact factorization. Both μ_i^+ and μ_i^- are convex combinations of the block value centroids, so their distance is at most the diameter in Eq. (38). This is the standard total-variation expectation argument Gibbs & Su (2002), specialized to attention on the selected blocks and retaining the exact value-discrepancy factor as in the Top-k total-variation theory Tzachristas et al. (2025) and the cumulant block-sparse analysis Tian et al. (2026). □

Complete candidate-set coverage implies $C = F$, hence $\eta_i^{\mathrm{mig}} = 0$ and $O_i^{\mathrm{CC}} = O_i^{\mathrm{FD}}$. When the score error and Full-Domain score gap satisfy the coverage condition, the two selections produce the same exact-computation set and the output error caused by changing the selected set is zero. The theorem quantifies the effect of replacing F with C. A block in $F \cap S_r$ outranks every block in $S_r \setminus F$ under the shared relevance-and-Top-K rule, so all blocks in $F \cap S_r$ are retained in C. Consequently, $F \setminus C = F \setminus S_r$. The always-included-block result above therefore yields

$$F \setminus C \subseteq \mathcal{A}_r^U, \qquad \frac{W_i(F \setminus C)}{W_i(F)} \le \frac{W_i(\mathcal{A}_r^U)}{W_i(F)}. \tag{40}$$

The second term in Eq. (33) records the exact mass of replacement blocks, while Eq. (38) records their value geometry. The partial-coverage error therefore depends jointly on replacement mass and value geometry, consistent with analyses separating attention importance from residual-approximation error Zhou et al. (2026).

J Numerical validation on captured attention tensors

We capture the Q/K/V tensors produced by QUASAR at diffusion step 34 and transformer layer 14 and reconstruct exact and recovered attention from these fixed inputs. The capture contains all 40 attention heads; we select one evenly spaced query row from each of the 256 query blocks, giving 10,240 rows per configuration. Draft uses its query-dependent candidate sets, while XAttention uses 156 of 256 candidate blocks. Q/K score products use float32; softmax reductions and bound evaluation use float64 and log-sum-exp stabilization. The comparison measures fixed-input approximation error relative to the BF16 generation path.

Evaluation relative to dense attention. We reconstruct all three selection configurations on the same 40 heads and 256 sampled query positions per head, using batched FP32 QK products with TF32 disabled and FP64 reductions. Table 30 reports the bound on error relative to dense attention, using B_i^{pair} within each selection domain. The composed upper bound holds for all 92,160 evaluated query–head/configuration pairs within the numerical tolerance $10^{-10}\max(1, B, E)$. At 79.69% sparsity, the Full-Domain relative Frobenius error decreases from 23.86% for exact-only attention to 22.37% for recovered attention, with an upper bound of 24.13%. The corresponding Draft and XAttention errors are 22.73% and 22.70%, bounded by 26.79% and 27.22% after including the outside-candidate contribution. These are attention-output errors on the sampled rows; generation quality is reported in the main experiments.

Table 29: **Recovery error and numerical bounds.**

(a) Decomposition residual and empirical improvement

Domain	Sparsity	Rows	Max residual	Rows above bound	Improved	Median gain
Full-Domain	84.77%	10,240	1.51×10^{-15}	0	95.44%	4.94%
Full-Domain	79.69%	10,240	1.40×10^{-15}	0	95.92%	4.50%
Full-Domain	75.00%	10,240	1.32×10^{-15}	0	96.29%	4.42%
Draft Candidate	84.77%	10,240	1.81×10^{-15}	0	93.31%	3.46%
Draft Candidate	79.69%	10,240	1.32×10^{-15}	0	93.28%	3.06%
Draft Candidate	75.00%	10,240	1.52×10^{-15}	0	93.12%	2.88%
XAttention Candidate	84.77%	10,240	1.63×10^{-15}	0	93.46%	3.74%
XAttention Candidate	79.69%	10,240	1.40×10^{-15}	0	93.50%	3.39%
XAttention Candidate	75.00%	10,240	1.34×10^{-15}	0	93.51%	3.30%

(b) Bound-to-error ratio

Domain	Sparsity	$B_i^{\mathrm{HCR}}/E_i^{\mathrm{rec}}$			$B_i^{\mathrm{pair}}/E_i^{\mathrm{rec}}$	$B_i^{\mathrm{hier}}/E_i^{\mathrm{rec}}$	$\Pr(B_i^{\mathrm{HCR}} < B_i^{\mathrm{hier}})$
		p50	p90	p95	p50	p50	
Full-Domain	84.77%	3.43	5.53	6.48	1.07	1.52	11.71%
Full-Domain	79.69%	3.64	5.92	6.98	1.06	1.46	7.83%
Full-Domain	75.00%	3.84	6.27	7.49	1.06	1.43	5.75%
Draft Candidate	84.77%	3.60	5.92	6.88	1.05	1.42	7.61%
Draft Candidate	79.69%	3.90	6.54	7.70	1.05	1.37	4.48%
Draft Candidate	75.00%	4.24	7.20	8.67	1.04	1.33	2.46%
XAttention Candidate	84.77%	3.59	5.89	6.84	1.06	1.45	8.00%
XAttention Candidate	79.69%	3.90	6.46	7.58	1.05	1.40	4.74%
XAttention Candidate	75.00%	4.22	7.01	8.35	1.05	1.37	2.78%

(c) Reference choice for the spectral bound and its sufficient condition

Domain	Sparsity	Forward	Reverse	Midpoint	Tie	Condition met	Improved if met
Full-Domain	84.77%	0.00%	37.06%	62.94%	0.00%	0.0195%	100%
Full-Domain	79.69%	0.00%	46.84%	53.16%	0.00%	0.0000%	-
Full-Domain	75.00%	0.00%	53.71%	46.29%	0.00%	0.0000%	-
Draft Candidate	84.77%	0.00%	44.57%	55.43%	0.00%	0.0000%	-
Draft Candidate	79.69%	0.00%	55.26%	44.74%	0.00%	0.0000%	-
Draft Candidate	75.00%	0.00%	62.11%	37.89%	0.00%	0.0000%	-
XAttention Candidate	84.77%	0.00%	43.68%	56.32%	0.00%	0.0000%	-
XAttention Candidate	79.69%	0.00%	54.48%	45.52%	0.00%	0.0000%	-
XAttention Candidate	75.00%	0.00%	61.60%	38.40%	0.00%	0.0000%	-

The pairwise triangle-inequality bound attains median bound-to-error ratios of 1.04–1.07 on these captured tensors. The spectral bound derived from the HCR inequality has median ratios of 3.43–4.24. In panel (c), two rows meet the sufficient condition, and both improve.

Within each KV domain, the sufficient condition based on the pairwise triangle-inequality bound holds for 32.38%–39.50% of all sampled rows and covers 34.70%–41.02% of the rows that actually improve. Relative to dense attention, the condition holds for 34.61%–39.50% of Full-Domain rows, 9.21%–9.70% of Draft rows, and 11.21%–11.72% of XAttention rows; the corresponding empirical improvement rates are 95.44%–96.29%, 96.02%–96.77%, and 95.86%–96.51%. The direct logit-range bound, evaluated with the valid diameter upper bound $2 \max_j \|v_j - \bar{v}\|_2$, likewise holds on these captured attention tensors.

K Hierarchical recovery proofs and auxiliary error bounds

K.1 Group-wise recovery definitions

For the following analysis, all exact quantities are defined over the same KV tiles $D_r = M_r \dot{\cup} R_r$. Let $D_r^+ = \{b \in D_r : n_b > 0\}$ and $R_r^+ = \{b \in R_r : n_b > 0\}$. Blocks with no valid token have zero contribution and are omitted. For every $b \in D_r^+$, define

$$Z_{i,b} = \sum_{j \in B_b \cap \mathcal{V}_r} e^{s_{ij}}, \qquad N_{i,b} = \sum_{j \in B_b \cap \mathcal{V}_r} e^{s_{ij}} v_j, \qquad o_{i,b} = \frac{N_{i,b}}{Z_{i,b}}, \tag{41}$$

Table 30: **Error and improvement bounds relative to dense attention.** Each configuration uses the same 10,240 sampled query/head rows. Relative errors and bounds are aggregate Frobenius ratios. The total bound uses B_i^{pair} plus the outside-candidate term; "Improved" is the fraction with smaller error relative to the dense output after recovery. The total bound holds for all 92,160 evaluated query–head/configuration pairs within the stated numerical tolerance.

Selection	Sparsity	Exact-only Rel.	Recovered Rel.	Bound Rel.	Condition met	Improved
Full-Domain	84.77%	28.91%	26.92%	29.24%	34.61%	95.44%
Full-Domain	79.69%	23.86%	22.37%	24.13%	36.81%	95.92%
Full-Domain	75.00%	20.64%	19.46%	20.85%	39.50%	96.29%
Draft Candidate	84.77%	28.91%	27.31%	32.20%	9.70%	96.02%
Draft Candidate	79.69%	23.85%	22.73%	26.79%	9.21%	96.42%
Draft Candidate	75.00%	20.63%	19.81%	23.28%	9.49%	96.77%
XAttention Candidate	84.77%	28.92%	27.27%	32.61%	11.21%	95.86%
XAttention Candidate	79.69%	23.86%	22.70%	27.22%	11.32%	96.30%
XAttention Candidate	75.00%	20.64%	19.77%	23.76%	11.72%	96.51%

Table 31: **Contributions to the hierarchical recovery bound.**

		Mean bound share			Median vector norm		
Domain	**Sparsity**	**Within**	**Between**	**Branch**	$\|x\|_2$	$\|y\|_2$	$\|z\|_2$
Full-Domain	84.77%	31.93%	5.73%	62.34%	0.1008	0.0239	1.1922
Full-Domain	79.69%	30.31%	5.32%	64.37%	0.0720	0.0170	0.9752
Full-Domain	75.00%	29.23%	5.03%	65.74%	0.0552	0.0129	0.8227
Draft Candidate	84.77%	29.81%	4.96%	65.23%	0.0751	0.0177	1.0376
Draft Candidate	79.69%	27.97%	4.52%	67.51%	0.0492	0.0116	0.7980
Draft Candidate	75.00%	26.63%	4.18%	69.17%	0.0342	0.0082	0.6288
XAttention Candidate	84.77%	30.56%	5.05%	64.40%	0.0801	0.0192	1.0439
XAttention Candidate	79.69%	28.93%	4.67%	66.41%	0.0540	0.0129	0.8045
XAttention Candidate	75.00%	27.84%	4.39%	67.77%	0.0390	0.0095	0.6391

where $s_{ij} = q_i^\top k_j/\sqrt{d}$. Define the exact selected and recovered states by

$$\begin{aligned} Z_i^M &= \sum_{b\in M_r\cap D_r^+} Z_{i,b}, \quad N_i^M = \sum_{b\in M_r\cap D_r^+} N_{i,b}, \quad O_i^M = \frac{N_i^M}{Z_i^M}, \\ Z_i^R &= \sum_{b\in R_r^+} Z_{i,b}, \quad O_i^R = \frac{1}{Z_i^R}\sum_{b\in R_r^+} N_{i,b}. \end{aligned} \tag{42}$$

The exact output on the selection domain is $O_i = (N_i^M + Z_i^R O_i^R)/(Z_i^M + Z_i^R)$. For each group define

$$\widehat{O}_{r,g}^R = \frac{\widehat{N}_{r,g}^R}{\widehat{Z}_{r,g}^R}, \qquad \widehat{\alpha}_{i,g} = \frac{\widehat{Z}_{r,g}^R}{Z_i^M + \widehat{Z}_{r,g}^R}, \qquad \alpha_i = \frac{Z_i^R}{Z_i^M + Z_i^R}. \tag{43}$$

For a fixed query, abbreviate $\widehat{O}_i^R = \widehat{O}_{r,g(i)}^R$ and $\widehat{\alpha}_i = \widehat{\alpha}_{i,g(i)}$. For each fixed query $i \in G_{r,g(i)}$, write $\widehat{Z}_i^R = \widehat{Z}_{r,g(i)}^R$ and $\widehat{N}_i^R = \widehat{N}_{r,g(i)}^R$. The exact-only reference is the identity

$$O_i - O_i^M = \alpha_i(O_i^R - O_i^M), \qquad E_i^{\text{exact}} := \|O_i - O_i^M\|_2 = \alpha_i \Delta_i^{RM}, \quad \Delta_i^{RM} := \|O_i^R - O_i^M\|_2. \tag{44}$$

We use the standard convention

$$\mathrm{TV}(P,Q) = \frac{1}{2}\sum_x |P(x) - Q(x)| = \sup_A |P(A) - Q(A)|. \tag{45}$$

For $b \in R_r^+$, let

$$\beta_{i,b} = \frac{Z_{i,b}}{Z_i^R}, \qquad \widehat{\beta}_{r,g,b} = \frac{\widehat{Z}_{r,g,b}^R}{\widehat{Z}_{r,g}^R}, \qquad \widehat{Z}_{r,g,b}^R = n_b e^{\rho_{r,g,b}^\Gamma}, \qquad \pi_{i,b,j} = \frac{e^{s_{ij}}}{Z_{i,b}}, \qquad \nu_{b,j} = \frac{1}{n_b}, \tag{46}$$

Table 32: **Changes in selected blocks and restriction to a candidate set.**

(a) Change between final exact-computation sets

Selection	Sparsity	Selection recall	Nonzero	$\overline{\eta}$	$\overline{\|\Delta O\|_2}$	$\overline{\eta\|\Delta\mu\|_2}$	Max residual
Draft Candidate	84.77%	0.999637	1.04%	0.000266	0.001652	0.001652	3.43×10^{-15}
Draft Candidate	79.69%	0.998201	5.58%	0.001315	0.007779	0.007779	3.54×10^{-15}
Draft Candidate	75.00%	0.993633	17.71%	0.003977	0.020648	0.020648	4.42×10^{-15}
XAttention Candidate	84.77%	0.999897	0.27%	0.000166	0.001324	0.001324	3.24×10^{-15}
XAttention Candidate	79.69%	0.999801	0.69%	0.000261	0.001977	0.001977	2.78×10^{-15}
XAttention Candidate	75.00%	0.999649	1.32%	0.000393	0.002486	0.002486	2.95×10^{-15}

(b) Observed relations

Selection	Sparsity	$\rho(\text{selection recall}, \eta)$	$\rho(\eta, \|\Delta O\|_2)$
Draft Candidate	84.77%	-0.750	0.923
Draft Candidate	79.69%	-0.660	0.896
Draft Candidate	75.00%	-0.661	0.917
XAttention Candidate	84.77%	-0.686	0.962
XAttention Candidate	79.69%	-0.656	0.952
XAttention Candidate	75.00%	-0.775	0.932

(c) Restriction from the legal KV-block set to a candidate set

Selection	Rows	Candidate fraction	Removed mass	Output error	Max residual
Draft Candidate	10,240	0.614348	0.097596	0.319566	5.70×10^{-15}
XAttention Candidate	10,240	0.609375	0.108211	0.335467	7.26×10^{-15}

Panel (a) compares Full-Domain and Candidate-Constrained selected sets using the same selection-recall definition as Table 2, on the captured tensors used for the selected-set comparison. Figure 5 uses the captured tensors on which the candidate scores were fit. Panel (c) compares the complete legal KV-block set with the candidate set. Correlations are computed over captured-tensor and query-block observations.

and define

$$\tau_{i,b}^{\mathrm{tok}} = \mathrm{TV}(\pi_{i,b}, \nu_b), \qquad \tau_{i,g}^{\mathrm{blk}} = \mathrm{TV}(\beta_i, \widehat{\beta}_{r,g}), \tag{47}$$

with $\bar{v}_b = \sum_j \nu_{b,j} v_j$, $D_b^V = \max_{j,k \in B_b \cap \mathcal{V}_r} \|v_j - v_k\|_2$, and $D_R^{\bar{V}} = \max_{b,c \in R_r^+} \|\bar{v}_b - \bar{v}_c\|_2$.

K.2 Pairwise triangle-inequality refinement

The bound applies to either KV domain, $D_r = \Omega_\Gamma$ or $D_r = S_r^{\mathrm{CC}}$, independently of candidate containment. To retain cancellation between terms in the hierarchical decomposition, let x_i, y_i, z_i denote the three vector terms in Eq. (10), and b_x, b_y, b_z their respective bounds in Eq. (11). Applying the triangle inequality after pairing two vector terms gives

$$B_i^{\mathrm{pair}} = \min\left\{\|x_i + y_i\|_2 + b_z,\ \|x_i + z_i\|_2 + b_y,\ \|y_i + z_i\|_2 + b_x\right\}. \tag{48}$$

The three pairings give $\|\widehat{O}_i - O_i\|_2 \le B_i^{\mathrm{pair}} \le b_x + b_y + b_z$. Appendix K.4 provides the expanded terms and treats empty or singleton recovered domains.

K.3 Conditional recovery bound from logit ranges

Let $\omega_{i,b}$ be the within-block centered-logit range, Ω_i^R the recovered-block log-mass-ratio range, and $\zeta_i^R = |\log(\widehat{Z}_{r,g(i)}^R / Z_i^R)|$. The bound relating total variation to Hilbert's projective metric Cohen & Fausti (2023) gives

$$\begin{aligned}\|\widehat{O}_i - O_i\|_2 \le \widehat{\alpha}_i \Bigg[\sum_{b \in R_r^+} \beta_{i,b} D_b^V \tanh\left(\frac{\omega_{i,b}}{4}\right) + D_R^{\bar{V}} \tanh\left(\frac{\Omega_i^R}{4}\right) \Bigg] \\ + \Delta_i^{RM} \tanh\left(\frac{\zeta_i^R}{4}\right).\end{aligned} \tag{49}$$

The bound is expressed in terms of within-block logit ranges, recovered-block log-mass-ratio ranges, branch-mass discrepancy, and value diameters. Appendix K.10 gives the derivation.

K.4 Proof of the hierarchical output error theorem

Fix an arbitrary query $i \in G_{r,g(i)}$ and use the state of its fixed recovery group. Assume $Z_i^M, Z_i^R, \widehat{Z}_{r,g(i)}^R > 0$. Since

$$O_i = (1-\alpha_i)O_i^M + \alpha_i O_i^R, \qquad \widehat{O}_i = (1-\widehat{\alpha}_i)O_i^M + \widehat{\alpha}_i \widehat{O}_{r,g(i)}^R,$$

subtracting gives

$$\widehat{O}_i - O_i = \widehat{\alpha}_i(\widehat{O}_{r,g(i)}^R - O_i^R) + (\widehat{\alpha}_i - \alpha_i)(O_i^R - O_i^M). \tag{50}$$

For $b \in R_r^+$, let $\bar{v}_b = \sum_j \nu_{b,j} v_j$. Then

$$\widehat{O}_{r,g(i)}^R - O_i^R = \sum_{b \in R_r^+} \beta_{i,b}(\bar{v}_b - o_{i,b}) + \sum_{b \in R_r^+} (\widehat{\beta}_{r,g(i),b} - \beta_{i,b})\bar{v}_b. \tag{51}$$

Combining Eqs. (50) and (51) proves the exact three-level identity in Theorem 2.

For the first sum, the expectation bound using total variation and diameter Gibbs & Su (2002), with the convention in Eq. (45), gives

$$\|\bar{v}_b - o_{i,b}\|_2 \le \tau_{i,b}^{\mathrm{tok}} D_b^V. \tag{52}$$

For the second sum, the same inequality applied to $b \mapsto \bar{v}_b$ gives

$$\left\| \sum_b (\widehat{\beta}_{r,g(i),b} - \beta_{i,b})\bar{v}_b \right\|_2 \le \tau_{i,g(i)}^{\mathrm{blk}} D_R^{\bar{V}}. \tag{53}$$

The triangle inequality proves Eq. (11).

If $R_r^+ = \varnothing$, the recovery state is $(-\infty, 0, 0)$ and the merged output is the exact selected-branch output, so the recovered-region error is absent. If $|R_r^+| = 1$, the between-block term is zero; constant values or uniform within-block logits likewise make the corresponding local distortion term zero. The theorem's positivity assumptions exclude only cases with zero normalization mass.

Expanded directional terms. The vectors used in Eq. (48) are

$$\begin{aligned} x_i &= \widehat{\alpha}_i \sum_{b \in R_r^+} \beta_{i,b}(\bar{v}_b - o_{i,b}), \\ y_i &= \widehat{\alpha}_i \sum_{b \in R_r^+} (\widehat{\beta}_{r,g(i),b} - \beta_{i,b})\bar{v}_b, \\ z_i &= (\widehat{\alpha}_i - \alpha_i)(O_i^R - O_i^M), \end{aligned} \tag{54}$$

K.5 Recovery versus exact-only attention

Define

$$\mathcal{C}_i^R = \sum_{b \in R_r^+} \beta_{i,b}\tau_{i,b}^{\mathrm{tok}} D_b^V + \tau_{i,g(i)}^{\mathrm{blk}} D_R^{\bar{V}}. \tag{55}$$

Improvement from recovery. The primary hierarchical bound and $E_i^{\mathrm{exact}} = \alpha_i \Delta_i^{RM}$ imply

$$\widehat{\alpha}_i \mathcal{C}_i^R + |\widehat{\alpha}_i - \alpha_i|\Delta_i^{RM} < \alpha_i \Delta_i^{RM} \quad \Longrightarrow \quad E_i^{\mathrm{rec}} < E_i^{\mathrm{exact}}. \tag{56}$$

The refined decomposition bound gives the sharper a-posteriori sufficient condition

$$B_i^{\mathrm{pair}} < E_i^{\mathrm{exact}} \quad \Longrightarrow \quad E_i^{\mathrm{rec}} < E_i^{\mathrm{exact}}. \tag{57}$$

If $\widehat{Z}_{r,g(i)}^R = Z_i^R$ and $\alpha_i > 0$, the comparison is exact:

$$E_i^{\mathrm{rec}} < E_i^{\mathrm{exact}} \quad \Longleftrightarrow \quad \|\widehat{O}_{r,g(i)}^R - O_i^R\|_2 < \Delta_i^{RM}. \tag{58}$$

Thus Eq. (56) follows from the hierarchical bound, Eq. (57) is the sharper condition obtained after pairing vector terms, and Eq. (58) is an iff statement only under exact aggregate recovery normalization.

The spectral bound derived from the HCR inequality also gives

$$B_i^{\mathrm{HCR}} < E_{i,D}^{\mathrm{exact}} \quad \Longrightarrow \quad E_{i,D}^{\mathrm{rec}} < E_{i,D}^{\mathrm{exact}}. \tag{59}$$

The exact-only identity in Eq. (44) gives $E_i^{\mathrm{exact}} = \alpha_i \Delta_i^{RM}$. Applying the primary hierarchical bound therefore proves the sufficient implication in Eq. (56). If $\widehat{Z}_{r,g(i)}^R = Z_i^R$, then $\widehat{\alpha}_i = \alpha_i$ and the exact branch identity reduces to

$$\widehat{O}_i - O_i = \alpha_i(\widehat{O}_{r,g(i)}^R - O_i^R).$$

For $\alpha_i > 0$, comparison with $E_i^{\mathrm{exact}} = \alpha_i \Delta_i^{RM}$ proves the iff statement; replacing the exact conditional error by its upper bound $\mathcal{C}_i^R$ gives the stated sufficient condition.

K.6 Spectral error bound from the Hammersley–Chapman–Robbins inequality

Define the exact distribution and the group-shared recovery distribution on the common valid-token domain by

$$P_i^D(j) = \frac{e^{s_{ij}}}{Z_i^M + Z_i^R}, \qquad Q_i^R(j) = \frac{e^{\widetilde{s}_{ij}}}{Z_i^M + \widehat{Z}_{r,g(i)}^R}, \qquad \widetilde{s}_{ij} = \begin{cases} s_{ij}, & b(j) \in M_r, \\ \rho_{r,g(i),b(j)}^{\Gamma}, & b(j) \in R_r^+. \end{cases} \tag{60}$$

Let $m_i = (P_i^D + Q_i^R)/2$ and

$$\mathfrak{T}_i = \sum_j \frac{(P_i^D(j) - Q_i^R(j))^2}{P_i^D(j) + Q_i^R(j)}, \qquad \chi^2(P_i^D \| m_i) = \chi^2(Q_i^R \| m_i) = \frac{\mathfrak{T}_i}{2}. \tag{61}$$

Theorem 4 (Spectral bound from the Hammersley–Chapman–Robbins inequality)**.** *For probability distributions P_i^D, Q_i^R on a common finite domain, let $\Sigma_P^V = \mathbb{E}_P[(V - \mathbb{E}_P V)(V - \mathbb{E}_P V)^\top]$ denote the value covariance under P. Then*

$$\begin{aligned} B_{i,Q}^{\mathrm{HCR}} &= \sqrt{\chi^2(P_i^D \| Q_i^R) \lambda_{\max}(\Sigma_{Q_i^R}^V)}, \qquad B_{i,P}^{\mathrm{HCR}} = \sqrt{\chi^2(Q_i^R \| P_i^D) \lambda_{\max}(\Sigma_{P_i^D}^V)}, \\ B_{i,M}^{\mathrm{HCR}} &= \sqrt{2\mathfrak{T}_i \lambda_{\max}(\Sigma_{m_i}^V)}, \qquad B_i^{\mathrm{HCR}} = \min\{B_{i,Q}^{\mathrm{HCR}}, B_{i,P}^{\mathrm{HCR}}, B_{i,M}^{\mathrm{HCR}}\}. \end{aligned} \tag{62}$$

Then $\|\widehat{O}_i - O_i^D\|_2 \le B_i^{\mathrm{HCR}}$. This independent spectral bound complements the primary hierarchical bound.

K.7 Proof of the spectral bound

For either ordered pair $(P, Q) = (P_i^D, Q_i^R)$ or (Q_i^R, P_i^D), restrict to the entries where both distributions are positive and write $\mu_Q = \mathbb{E}_Q[V]$. Define a centered value matrix A_Q and a discrepancy vector r_Q by their rows and entries:

$$A_Q(j,:) = \sqrt{Q(j)}(v_j - \mu_Q)^\top, \qquad r_Q(j) = \frac{P(j) - Q(j)}{\sqrt{Q(j)}}. \tag{63}$$

Normalization of P, Q gives the exact identities

$$\begin{aligned} \mathbb{E}_P[V] - \mathbb{E}_Q[V] &= A_Q^\top r_Q, \\ A_Q^\top A_Q &= \Sigma_Q^V, \qquad \|r_Q\|_2^2 = \chi^2(P\|Q). \end{aligned} \tag{64}$$

A single operator-norm inequality, equivalent to applying the Hammersley–Chapman–Robbins inequality and Cauchy–Schwarz to vector-valued expectations, gives

$$\|A_Q^\top r_Q\|_2 \le \|A_Q^\top\|_{2\to 2} \|r_Q\|_2 = \sqrt{\lambda_{\max}(\Sigma_Q^V) \chi^2(P\|Q)}. \tag{65}$$

This proves the forward and reverse bounds without separately bounding the three recovery error terms. For $m = (P_i^D + Q_i^R)/2$, the expectation difference between P_i^D and Q_i^R is twice that between P_i^D and m, while $\chi^2(P_i^D \| m) = \mathfrak{T}_i/2$; applying the same argument yields $B_{i,M}^{\mathrm{HCR}}$. The minimum of the three upper bounds proves Theorem 4. Tightness depends on the alignment of r_Q with the leading singular subspace of $A_Q^\top$.

K.8 COARSE WORST-CASE RECOVERY ERROR BOUND

Fix a query $i \in G_{r,g(i)}$. All exact quantities below use the same KV tiles $D_r = M_r \dot\cup R_r$. Assume $R_r^+ \neq \varnothing$ and write, for $b \in R_r^+$,

$$s_{ij} = \rho^{\Gamma}_{r,g(i),b} + \delta_{ij}, \qquad |\delta_{ij}| \le \varepsilon_{i,b}. \tag{66}$$

Let

$$\bar v_b = \frac{1}{n_b} \sum_{j \in B_b \cap \mathcal{V}_r} v_j, \qquad \eta_j = v_j - \bar v_b, \qquad D_b = \frac{1}{n_b} \sum_{j \in B_b \cap \mathcal{V}_r} \|\eta_j\|_2. \tag{67}$$

Define $Z_{i,b}, \widehat{Z}^R_{r,g(i),b}, N_{i,b}, \widehat{N}^R_{r,g(i),b}$ as in Eq. (41) and the recovery definitions. Let Z_i, N_i, O_i be the exact quantities on the selection domain and set

$$p_{i,b} = \frac{Z_{i,b}}{Z_i}, \qquad p_i^R = \sum_{b \in R_r^+} p_{i,b}, \qquad d_{i,b} = e^{\varepsilon_{i,b}} - 1. \tag{68}$$

We state the following coarse worst-case recovery bound:

Theorem 5 (Coarse worst-case recovery error bound)**.**

$$\|\widehat{O}_i - O_i\|_2 \le \frac{\displaystyle\sum_{b \in R_r^+} p_{i,b} d_{i,b} \left(\|\bar v_b - O_i\|_2 + e^{\varepsilon_{i,b}} D_b\right)}{\displaystyle 1 - p_i^R + \sum_{b \in R_r^+} p_{i,b} e^{-\varepsilon_{i,b}}}. \tag{69}$$

In particular, with $\varepsilon_i^R = \max_b \varepsilon_{i,b}$, $D_i^R = \max_b D_b$, *and* $C^R_{i,\infty} = \max_b \|\bar v_b - O_i\|_2$,

$$\|\widehat{O}_i - O_i\|_2 \le \frac{p_i^R (e^{\varepsilon_i^R} - 1)(C^R_{i,\infty} + e^{\varepsilon_i^R} D_i^R)}{1 - p_i^R + p_i^R e^{-\varepsilon_i^R}}. \tag{70}$$

Proof. For each recovered block, $e^{-\varepsilon_{i,b}} \widehat{Z}^R_{r,g(i),b} \le Z_{i,b} \le e^{\varepsilon_{i,b}} \widehat{Z}^R_{r,g(i),b}$, hence $|\widehat{Z}^R_{r,g(i),b} - Z_{i,b}| \le d_{i,b} Z_{i,b}$. Writing $v_j = \bar v_b + \eta_j$ and using $\sum_j \eta_j = 0$ yields

$$\|(\widehat{N}^R_{r,g(i),b} - N_{i,b}) - O_i(\widehat{Z}^R_{r,g(i),b} - Z_{i,b})\|_2 \le Z_{i,b} d_{i,b} (\|\bar v_b - O_i\|_2 + e^{\varepsilon_{i,b}} D_b).$$

Summing and dividing by Z_i gives the numerator. Moreover,

$$\frac{Z_i + \Delta Z_i}{Z_i} \ge 1 - p_i^R + \sum_{b \in R_r^+} p_{i,b} e^{-\varepsilon_{i,b}}.$$

The exact perturbation identity in Eq. (74) then proves the detailed bound, and taking maxima proves the compact bound. □

Group-wise recovery specialization. For the fixed query above, the approximate block states in this bound are $\widehat{Z}^R_{r,g(i),b} = n_b e^{\rho^{\Gamma}_{r,g(i),b}}$ and $\widehat{N}^R_{r,g(i),b} = e^{\rho^{\Gamma}_{r,g(i),b}} V_b^{\Sigma}$. Thus the detailed and compact bounds apply separately to each group, with $\varepsilon_{i,b}$ measured relative to the group score. When $G_{\mathrm{rec}} = 1$, $\rho = a$ and they reduce to the single-group specialization.

For the exact decomposition vectors in Eq. (54), write $e_i = \widehat{O}_i - O_i = x_i + y_i + z_i$. By the scalar estimates in Eq. (11), set $b_x = \widehat{\alpha}_i \sum_b \beta_{i,b} \tau^{\mathrm{tok}}_{i,b} D_b^V$, $b_y = \widehat{\alpha}_i \tau^{\mathrm{blk}}_{i,g(i)} D_R^V$, and $b_z = |\widehat{\alpha}_i - \alpha_i| \Delta_i^{RM}$, so that $\|x_i\|_2 \le b_x$, $\|y_i\|_2 \le b_y$, and $\|z_i\|_2 \le b_z$. Applying the triangle inequality to each pairing gives

$$\|e_i\|_2 = \|(x_i + y_i) + z_i\|_2 \le \|x_i + y_i\|_2 + b_z, \tag{71}$$

$$\|e_i\|_2 = \|(x_i + z_i) + y_i\|_2 \le \|x_i + z_i\|_2 + b_y, \tag{72}$$

$$\|e_i\|_2 = \|(y_i + z_i) + x_i\|_2 \le \|y_i + z_i\|_2 + b_x. \tag{73}$$

Taking the minimum proves Eq. (48). The argument applies separately to each fixed query and recovery group.

Let N_i^R, Z_i^R be the exact recovered-region contributions, and define $\Delta N_i = \widehat{N}^R_{r,g(i)} - N_i^R$ and $\Delta Z_i = \widehat{Z}^R_{r,g(i)} - Z_i^R$. If $Z_i + \Delta Z_i > 0$, then

$$\widehat{O}_i - O_i = \frac{\Delta N_i - O_i \Delta Z_i}{Z_i + \Delta Z_i}. \tag{74}$$

This algebraic identity is equivalent to the branch-level decomposition and provides a direct check of the normalization terms.

Proof. Substitute $\widehat{O}_i = (N_i + \Delta N_i)/(Z_i + \Delta Z_i)$ and $O_i = N_i/Z_i$ and simplify. □

K.9 Effects of query and key variation

For $i \in G_r$ and $j \in B_b \cap \mathcal{V}_r$, define

$$\begin{aligned} \bar{k}^{\mathcal{V}}_{r,b} &= \frac{1}{n_b} \sum_{j \in B_b \cap \mathcal{V}_r} k_j, & \mu^{Q,G}_{i,b} &= \frac{(q_i - \bar{q}_{r,g(i)})^\top \bar{k}_b}{\sqrt{d}}, \\ \nu^{\mathrm{val}}_{i,b} &= \frac{q_i^\top (\bar{k}^{\mathcal{V}}_{r,b} - \bar{k}_b)}{\sqrt{d}}, & \xi_{i,bj} &= \frac{q_i^\top (k_j - \bar{k}^{\mathcal{V}}_{r,b})}{\sqrt{d}}. \end{aligned} \tag{75}$$

The centroid over valid tokens is defined only for $n_b > 0$. Consequently, $\mathbb{E}_{u_b} \xi_{i,bj} = 0$, where u_b is uniform on $B_b \cap \mathcal{V}_r$, and the token logit admits the exact decomposition

$$s_{ij} = \rho^{\Gamma}_{r,g(i),b} + \mu^{Q,G}_{i,b} + \nu^{\mathrm{val}}_{i,b} + \xi_{i,bj}, \qquad m_{i,b} := \log \frac{Z_{i,b}}{\widehat{Z}^R_{r,g(i),b}} = \mu^{Q,G}_{i,b} + \nu^{\mathrm{val}}_{i,b} + \psi_{i,b}, \tag{76}$$

where $\psi_{i,b} = \log \mathbb{E}_{u_b} e^{\xi_{i,bj}}$. Since $\mathbb{E}_{u_b} \xi_{i,bj} = 0$, Jensen and Hoeffding's lemma Hoeffding (1963) yield

$$0 \le \psi_{i,b} \le \frac{\omega^2_{i,b}}{8}, \qquad \omega_{i,b} = \max_{j \in B_b \cap \mathcal{V}_r} \xi_{i,bj} - \min_{j \in B_b \cap \mathcal{V}_r} \xi_{i,bj}. \tag{77}$$

Thus sharing a query representation within the group contributes $\mu^{Q,G}_{i,b}$, valid-token mismatch contributes $\nu^{\mathrm{val}}_{i,b}$, and key variation among valid tokens in the block contributes the term $\psi_{i,b}$, whose upper bound depends on the within-block logit range.

K.10 Conditional bound independent of the number of recovered blocks

For the distributions in Theorem 2, the sharp bound relating total variation to Hilbert's projective metric Cohen & Fausti (2023) under the convention in Eq. (45) gives

$$\tau^{\mathrm{tok}}_{i,b} \le \tanh(\omega_{i,b}/4), \qquad \tau^{\mathrm{blk}}_{i,g(i)} \le \tanh(\Omega^R_i/4), \qquad |\widehat{\alpha}_i - \alpha_i| \le \tanh(\zeta^R_i/4),$$

where

$$\Omega^R_i = \max_{b \in R^+_r} m_{i,b} - \min_{b \in R^+_r} m_{i,b}, \qquad \zeta^R_i = \left| \log \frac{\widehat{Z}^R_{r,g(i)}}{Z^R_i} \right|.$$

Substitution into Eq. (11) proves Eq. (49). The TV convention is essential: the corresponding ℓ_1 inequality has an additional factor of two.

K.11 Variance-sensitive refinement

Let $\sigma^2_{V,b} = \mathbb{E}_{u_b} \|v_j - \bar{v}_b\|_2^2$. Change of measure and Cauchy–Schwarz give

$$\|o_{i,b} - \bar{v}_b\|_2 \le \sqrt{\chi^2(\pi_{i,b} \| u_b)}\, \sigma_{V,b}. \tag{78}$$

This refinement is an alternative local bound.

K.12 Recovery with a restricted candidate set

For Candidate-Constrained recovery, write $S = S_r^{\mathrm{CC}}$, let $U_r^{\mathrm{out}} = \Omega_\Gamma \setminus S$, and let O_i^S, O_i^U be exact outputs on the two domains. If $\lambda_i^S = Z_i^S / Z_i^\Omega$, then

$$\|\widehat{O}_i^{\mathrm{CC}} - O_i^\Omega\|_2 \le \|\widehat{O}_i^{\mathrm{CC}} - O_i^S\|_2 + (1-\lambda_i^S)\|O_i^S - O_i^U\|_2. \tag{79}$$

For Candidate-Constrained recovery, the hierarchical bound applies within the KV tiles, and Eq. (79) accounts for the contribution outside the candidate set.

Corollary 6 (Bounds relative to dense attention)**.** *Assume a nonempty set of valid tokens and the corresponding hypotheses of the recovery bounds. For either selection domain, let*

$$B_i^{\mathrm{tot}} = \begin{cases} B_i^{\mathrm{hier}}(\Omega_\Gamma), & \textit{Full-Domain}, \\ B_i^{\mathrm{hier}}(S) + (1-\lambda_i^S)\|O_i^S - O_i^U\|_2, & \textit{Candidate}, \end{cases} \tag{80}$$

with the outside term defined as zero if U is empty. Then $\|\widehat{O}_i - O_i^\Omega\|_2 \le B_i^{\mathrm{tot}}$. For any collection I of query/head rows with $\sum_{i\in I}\|O_i^\Omega\|_2^2 > 0$, stack the corresponding outputs into matrices. Their relative Frobenius error satisfies

$$\frac{\|\widehat{O}_I - O_I^\Omega\|_F}{\|O_I^\Omega\|_F} \le \frac{\sqrt{\sum_{i\in I}(B_i^{\mathrm{tot}})^2}}{\sqrt{\sum_{i\in I}\|O_i^\Omega\|_2^2}}. \tag{81}$$

For the exact output O_i^M on the same exact-computation set, the following condition guarantees that recovery reduces error relative to dense attention:

$$B_i^{\mathrm{tot}} < \|O_i^M - O_i^\Omega\|_2 \quad \Longrightarrow \quad \|\widehat{O}_i - O_i^\Omega\|_2 < \|O_i^M - O_i^\Omega\|_2. \tag{82}$$

Proof. The aggregate comparison uses the primary hierarchical bound; the directional and spectral quantities shown here are auxiliary bounds. Candidate-Constrained uses the exact identity $O_i^\Omega = \lambda_i^S O_i^S + (1-\lambda_i^S)O_i^U$ and the triangle inequality. Full-Domain has no domain-restriction term. Squaring the nonnegative row-wise bounds, summing, taking square roots, and dividing by the positive reference norm proves Eq. (81). Comparing the absolute bound with the selected-set error proves Eq. (82). □

The aggregate relative bound is a ratio of sums of squares on the sampled query/head rows. Relative errors are reported for rows with a positive reference norm. For a numerical backend output O_i^{impl}, any bound $\epsilon_i^{\mathrm{num}} \ge \|O_i^{\mathrm{impl}} - \widehat{O}_i\|_2$ gives

$$\|O_i^{\mathrm{impl}} - O_i^\Omega\|_2 \le B_i^{\mathrm{tot}} + \epsilon_i^{\mathrm{num}}. \tag{83}$$

The numerical term composes with the approximation bound, and the same sum-of-squares construction applies after it is included. All outputs here are evaluated before the subsequent output projection and network layers.

K.13 A direct logit-range error bound

The explicit distribution in Eq. (60) also gives a bound expressed through logit ranges, without exponential growth in the maximum logit deviation. Let $\widetilde{s}_{ij} = s_{ij}$ on selected blocks and $\widetilde{s}_{ij} = \rho^\Gamma_{r,g(i),b}$ on recovered blocks. For $D = D_r$, let $\mathcal{X}_{i,D} = \bigcup_{b\in D}(B_b \cap \mathcal{V}_r)$ be the valid-token set. Define

$$\delta_{ij} = s_{ij} - \widetilde{s}_{ij}, \qquad h_i = \max_j \delta_{ij} - \min_j \delta_{ij}, \qquad D_{V,D} = \max_{j,k\in\mathcal{X}_{i,D}} \|v_j - v_k\|_2. \tag{84}$$

Corollary 7 (Absolute error bound from logit ranges)**.** *For finite logits on the common nonempty valid domain,*

$$\|\widehat{O}_i - O_i^D\|_2 \le B_i^{\mathrm{logit}}(D) := D_{V,D}\tanh(h_i/4) \le D_{V,D}\tanh(\varepsilon_i/2) \tag{85}$$

whenever $\varepsilon_i \ge 0$ and $|s_{ij} - \rho^\Gamma_{r,g(i),b}| \le \varepsilon_i$ on all recovered tokens. The last bound also holds with any valid upper bound $\overline{D}_{V,D}$ on $D_{V,D}$. If $0 < \tau < \overline{D}_{V,D}$, the explicit condition

$$\varepsilon_i \le 2\,\mathrm{arctanh}(\tau/\overline{D}_{V,D}) \quad \Longrightarrow \quad \|\widehat{O}_i - O_i^D\|_2 \le \tau \tag{86}$$

is sufficient for the requested absolute tolerance.

Proof. The log probability ratio is $\log(P_i^D(j)/Q_i^R(j)) = \delta_{ij} + \log((Z_i^M + \widehat{Z}_{r,g(i)}^R)/(Z_i^M + Z_i^R))$. The normalization term is constant in j, so the log probability ratio has range h_i. Apply the bound relating total variation to Hilbert's projective metric Cohen & Fausti (2023), followed by the expectation bound using total variation and diameter. Selected tokens have $\delta_{ij} = 0$, and recovered tokens satisfy $|\delta_{ij}| \le \varepsilon_i$, giving $h_i \le 2\varepsilon_i$. Monotonicity of $\tanh$ proves the tolerance implication. □

One choice that avoids enumerating all query–key logits is

$$r_b^K = \max_{j \in B_b \cap \mathcal{V}_r} \|k_j - \bar{k}_b\|_2,$$
$$\varepsilon_i = \max_{b \in R_r^+} \frac{\|q_i - \bar{q}_{r,g(i)}\|_2 \|\bar{k}_b\|_2 + \|q_i\|_2 r_b^K}{\sqrt{d}}. \tag{87}$$

This follows by writing $q_i^\top k_j - \bar{q}_{r,g(i)}^\top \bar{k}_b = (q_i - \bar{q}_{r,g(i)})^\top \bar{k}_b + q_i^\top (k_j - \bar{k}_b)$ and applying Cauchy–Schwarz. The first term measures query sharing and vanishes for per-query recovery; the second measures within-block key variation. Both terms are computed from input summaries. The bound separates the effects of refining query groups and refining KV blocks, providing an input-based criterion for studying adaptive recovery granularity. Similarly, $2 \max_j \|v_j - c\|_2$ bounds the value diameter for any fixed center c. Empty recovery gives $h_i = \varepsilon_i = 0$ and zero error on the selected KV-block set. For Candidate-Constrained recovery, Eq. (79) adds the outside-candidate contribution to obtain the dense-reference bound.

L Online-softmax state merge

Equivalently, with $O_i^M = N_i^M / Z_i^M$ and $\widehat{U}_{r,g}^R = \widehat{N}_{r,g}^R / \widehat{Z}_{r,g}^R$ when the corresponding masses are positive,

$$\widehat{O}_i = O_i^M + \widehat{\alpha}_{i,g(i)} \big(\widehat{U}_{r,g(i)}^R - O_i^M \big), \qquad \widehat{\alpha}_{i,g} = \frac{\widehat{Z}_{r,g}^R}{Z_i^M + \widehat{Z}_{r,g}^R}. \tag{88}$$

This identity makes the amortization explicit: the recovery state is shared at group level, whereas the exact output, exact normalizer, and final mixture weight remain query-specific.

An online-softmax state $P = (m, l, c)$ represents the unnormalized denominator and numerator as $Z = e^m l$ and $N = e^m c$. Given two states over disjoint token or block partitions, let $m = \max(m_1, m_2)$ and define

$$l = e^{m_1 - m} l_1 + e^{m_2 - m} l_2, \qquad c = e^{m_1 - m} c_1 + e^{m_2 - m} c_2. \tag{89}$$

Then $e^m l = e^{m_1} l_1 + e^{m_2} l_2 = Z_1 + Z_2$ and likewise $e^m c = N_1 + N_2$. Therefore $c/l = (N_1 + N_2)/(Z_1 + Z_2)$, proving the partition-merge identity and the output/LSE conversion used by recovery. The construction is associative because it represents ordinary addition of unnormalized states. It requires the exact LSE and recovery logits to use the same natural-exponential score scale, and it requires both n_b and V_b^Σ to use the same token-validity set.