QUERY-GUIDED COARSE-TO-FINE KV CACHE COMPRESSION FOR UNIFIED MULTIMODAL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unified multimodal models have demonstrated strong capabilities in visual generation and editing, but long multimodal KV caches incur substantial memory and attention costs during generation. We propose a coarse-to-fine KV cache compression framework that progressively prunes redundant KV tokens while controlling the cost of importance estimation. The coarse stage exploits the local similarity of visual queries and their similar preferences over cached tokens. We sample one representative query from each local group to reduce the QK computation required for importance estimation. For stronger compression, the fine stage revisits the shortened cache at a later denoising step using updated, sampled queries. It ranks feature channels by their mean absolute query activations and uses the highest-scoring half for importance estimation, limiting the additional refinement cost without reducing retained KV dimensions. The evaluated pruning schedules improve the quality–cache trade-off over aggressive early pruning. On MagicBrush and EMU-Edit, refinement removes more than one fifth of the coarse cache with relative CLIP-I and DINO decreases below 1%. Separate systems measurements show approximately halved KV storage and 29.3–31.8% less attention-and-scoring computation, with workload-dependent latency gains.

1 INTRODUCTION

Unified multimodal models integrate visual understanding and generation within a single architecture (Deng et al., 2025; Liu et al., 2025; Wu et al., 2025a;b; Xie et al., 2024; 2025; Zhao et al., 2025). Their capabilities in text-to-image generation and instruction-guided image editing make efficient multimodal inference an important research problem.

Long conditioning caches are a recurring source of inference cost in cache-based multimodal generation. Text and reference-image representations, including visual features for understanding and generation, form a context whose keys and values are reused across denoising steps. Queries from the current image being denoised repeatedly attend to this context, so a longer cache increases both persistent storage and attention computation throughout generation.

Existing efficiency methods address complementary parts of this cost. Importance-based KV compression identifies less important tokens from query–key interactions or attention statistics (Zhang et al., 2023b; Li et al., 2024; Zhou et al., 2025; Feng et al., 2025; Wan et al., 2024; Tu et al., 2025; Zeng et al., 2026). However, directly estimating importance with all current queries and all feature channels can be costly when the historical cache is long. This scoring overhead can offset the savings from subsequent cache pruning. Other methods accelerate diffusion and flow-based generation by reducing the number of sampling steps (Song et al., 2023; Luo et al., 2023; Yin et al., 2024b;a; Frans et al., 2025; Geng et al., 2025) or reusing intermediate features (Liu et al., 2024). These approaches address different parts of the generation cost. Our focus is to shorten the KV cache without introducing excessive importance-estimation overhead.

We propose a query-guided coarse-to-fine KV cache compression framework. The coarse stage exploits the local similarity of visual queries and their preferences over cached tokens. Motivated by this, we partition visual queries into $n \times n$ spatial groups and sample one representative from each group. This reduces the scored query count to approximately $1/n^2$ of the original. When a tighter cache budget is required, the fine stage reassesses the shortened cache with queries sampled at a later denoising step. Query-guided half-channel scoring further limits refinement cost by using only

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.

the half of channels with the largest mean absolute query activations. Both stages reduce scoring computation without reducing the feature dimensions of the retained KV cache.

Experiments show that on MagicBrush and EMU-Edit, adding refinement to coarse pruning removes a further 21.1–21.3% of the remaining cache, with relative CLIP-I and DINO decrease below 1%. At tighter budgets, coarse-to-fine KV cache compression achieves better scores on all reported metrics while retaining fewer entries than coarse-only pruning. These results support a central design principle: preserve a sufficiently broad candidate cache early, then use later-step reassessment to make a more selective reduction (Secs. 3.3 and 3.4).

Our contributions are summarized as follows.

- We introduce a coarse-to-fine KV cache compression framework for unified multimodal generation. It progressively prunes historical conditioning KV tokens while controlling importance-estimation overhead.
- We design query-guided importance estimators: spatial query sampling and half-channel refinement. These two methods cut computational cost by a factor of n^2 and 2, respectively.
- Experiments on three benchmarks show that coarse-to-fine pruning improves the attainable quality–cache trade-off over aggressive coarse pruning.

2 METHOD

2.1 PROBLEM FORMULATION

We compress the reusable conditioning KV cache, not the current visual-token sequence. Text and reference-image representations provide the conditioning context, whose keys and values are cached across denoising steps. Queries from the current visual tokens attend jointly to these cached keys and the current visual keys. We refer to the conditioning cache as the *historical cache* to distinguish it from current-step keys and values.

For one attention layer and one conditioning branch, let $Q \in \mathbb{R}^{T\times H_q\times d}$ denote the current queries and $K^{\mathrm{cur}} \in \mathbb{R}^{T\times H_{kv}\times d}$ the current keys. The historical cache is $K^{\mathrm{hist}}, V^{\mathrm{hist}} \in \mathbb{R}^{N\times H_{kv}\times d}$, where T and N are the query and cache lengths. Under grouped-query attention, each KV head is shared by $G = H_q/H_{kv}$ query heads. We write query components as $Q_{i,h,g,c}$, with KV-head index h, associated query-head index g, and feature-channel index c. Layer, branch, and denoising-step indices are omitted for clarity.

2.2 FRAMEWORK OVERVIEW

Our framework first reduces the importance-estimation workload and then optionally imposes a tighter cache budget (Fig. 1). Coarse pruning uses locally sampled queries and corresponding current keys to score every historical candidate. Fine refinement reassesses the surviving candidates at a later denoising step using updated, sampled queries and selected feature channels. When further compression is unnecessary, generation directly reuses the coarse cache.

Both stages apply cumulative-importance selection and construct a dense cache with equal lengths across KV heads. Here, *coarse-to-fine* denotes progressive candidate reduction and later-step reassessment, rather than a return to exact importance scoring. Sampling and channel selection are used only to estimate importance: subsequent generation uses all current visual queries and the retained KV entries at their original feature dimensions.

2.3 KV IMPORTANCE ESTIMATION AND SELECTION

A shared importance estimator. We express both stages using a query-index set $\mathcal{I} \subseteq \{1,\dots,T\}$, a current-key index set $\mathcal{J} \subseteq \{1,\dots,T\}$, and a channel set $\mathcal{C}_h \subseteq \{1,\dots,d\}$ for each KV head. The

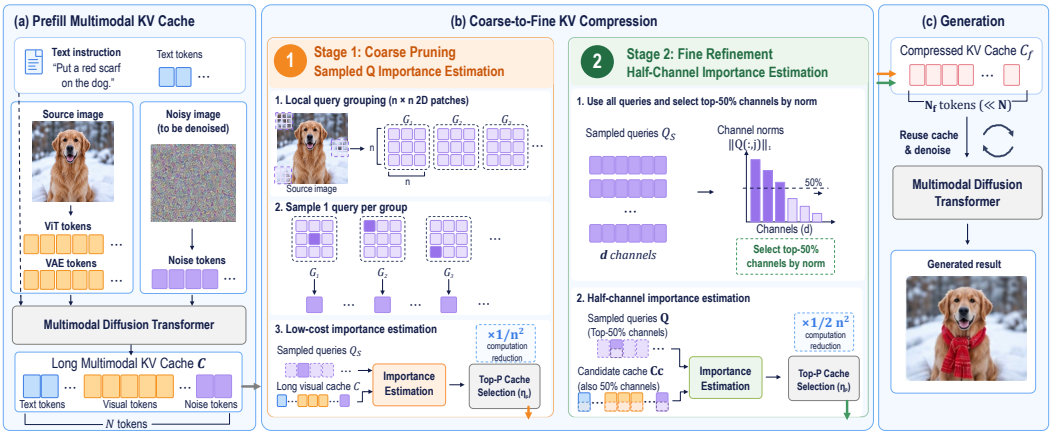

Figure 1: Coarse-to-fine compression of the conditioning KV cache. Coarse scoring samples queries and their corresponding current keys while evaluating all historical KV tokens. Optional fine refinement uses later sampled queries, all current keys, and half of the feature channels, selected by mean absolute query activation. Each stage applies Top-p selection and head refill. Generation then reuses the resulting cache without reducing the feature dimensions of its keys or values.

historical and current-key logits are

$$z^{\text{hist}}_{i,h,g,j} = \frac{1}{\sqrt{d}} \sum_{c \in \mathcal{C}_h} Q_{i,h,g,c} K^{\text{hist}}_{j,h,c}, \tag{1}$$

$$z^{\text{cur}}_{i,h,g,u} = \frac{1}{\sqrt{d}} \sum_{c \in \mathcal{C}_h} Q_{i,h,g,c} K^{\text{cur}}_{u,h,c}. \tag{2}$$

For each selected query, normalization includes all historical keys and the current keys indexed by $\mathcal{J}$:

$$Z_{i,h,g} = \sum_{v=1}^{N} \exp\left(z^{\text{hist}}_{i,h,g,v}\right) + \sum_{u \in \mathcal{J}} \exp\left(z^{\text{cur}}_{i,h,g,u}\right). \tag{3}$$

We average the normalized historical attention weights over the selected queries and the query heads associated with KV head h:

$$s_{h,j}(\mathcal{I}, \mathcal{J}, \mathcal{C}_h) = \frac{1}{|\mathcal{I}|G} \sum_{i \in \mathcal{I}} \sum_{g=1}^{G} \frac{\exp\left(z^{\text{hist}}_{i,h,g,j}\right)}{Z_{i,h,g}}. \tag{4}$$

Only historical head–token entries receive pruning scores. Current keys contribute to normalization but are not pruning candidates; values are not used in scoring. The full estimator uses all query positions, all current keys, and all feature channels.

Top-p selection. Selection is performed independently within each layer and conditioning branch, over all historical head–token pairs. We flatten the scores as $\mathbf{s}_h = [s_{h,1}, \ldots, s_{h,N}]^\top$ and $\mathbf{S} = [\mathbf{s}_1; \ldots; \mathbf{s}_{H_{kv}}] \in \mathbb{R}^{H_{kv}N}$. Let π be a descending ordering of $\mathbf{S}$, with ties resolved consistently in selection and refill. For a retained-importance threshold $p \in (0, 1]$, we select the shortest prefix satisfying

$$m^* = \min \left\{ m \in \{1, \ldots, H_{kv}N\} : \sum_{r=1}^{m} S_{\pi_r} \geq p \sum_{h=1}^{H_{kv}} \sum_{j=1}^{N} s_{h,j} \right\}. \tag{5}$$

Thus, p specifies the fraction of estimated historical importance mass covered by the initial selection, not a token-retention ratio. We denote selection followed by dense cache construction as

$$(\bar{K}, \bar{V}) = \mathcal{P}_p\left(K^{\text{hist}}, V^{\text{hist}}; \mathbf{S}\right). \tag{6}$$

The operator preserves head identities and may retain additional entries through the head-refill procedure in Sec. 2.6.

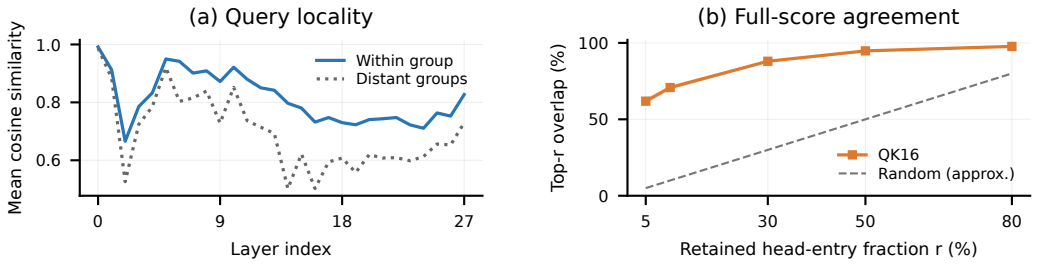

Figure 2: Evidence for local query sampling. (a) Mean query cosine similarity across layers for pairs within and across local groups. (b) Selection overlap with the full estimator at fixed image-KV head–token retention fractions. The diagnostic uses non-overlapping 4×4 spatial groups. The plotted fraction r corresponds to the diagnostic budget β in Sec. 3.5, not to the Top-p threshold.

2.4 Sampled-QK Importance Estimation for Coarse Pruning

Local query redundancy motivates scoring the cache with fewer query positions. The diagnostic in Fig. 2(a) shows higher similarity within local groups than across groups. For two queries q_i, q_j and a cached key k,

$$\frac{|q_i^\top k - q_j^\top k|}{\sqrt{d}} \leq \frac{\|q_i - q_j\|_2 \|k\|_2}{\sqrt{d}}. \tag{7}$$

Queries close in feature space therefore produce similar logits for keys with bounded norms. Cosine similarity supports this motivation when query norms are comparable, although it does not guarantee identical importance rankings. The selection overlap in Fig. 2(b) provides complementary evidence for sampled scoring.

We partition the visual query grid into non-overlapping $n \times n$ spatial groups. Let $\mathcal{G}_r$ contain the n^2 sequence indices in group r. One representative is sampled uniformly from each complete group:

$$i_r \sim \mathrm{Uniform}(\mathcal{G}_r), \qquad \mathcal{I}_n = \mathcal{B} \cup \{i_r\}_{r=1}^{R}, \tag{8}$$

where $\mathcal{B}$ contains query positions outside complete spatial groups, which are retained in the scoring set. The same sampled indices are shared across attention heads and used to gather corresponding current-step keys. All historical keys and all feature channels remain available for scoring:

$$s_{h,j}^{(1)} = s_{h,j}(\mathcal{I}_n, \mathcal{I}_n, \{1, \ldots, d\}). \tag{9}$$

Ignoring boundary effects, $|\mathcal{I}_n| \approx T/n^2$. Current-key sampling also shortens the normalization domain, but does not remove current visual tokens from generation. Because both the query average and the normalizer are sampled, this estimator approximates rather than reproduces full importance scores. A conservative threshold p_1 retains candidates for subsequent generation or optional refinement.

2.5 Query-Guided Channel Selection for Fine Refinement

Fine refinement supports an additional pruning decision without returning to full-query scoring. At a later denoising step, the *QChan50* estimator combines updated, locally sampled queries with query-guided half-channel scoring on the shortened historical cache.

Selecting scoring channels. For each KV head, we rank channels by their mean absolute query activation. Let $\mathcal{I}_f$ be sampled at the refinement step using the same spatial rule and group size as in coarse pruning. We compute

$$a_{h,c} = \frac{1}{|\mathcal{I}_f| G} \sum_{i \in \mathcal{I}_f} \sum_{g=1}^{G} |Q_{i,h,g,c}|, \tag{10}$$

and select the indices of the largest activations:

$$\mathcal{C}_h = \mathrm{TopChannels}_{d'}(a_{h,:}), \qquad d' = \left\lceil \frac{d}{2} \right\rceil. \tag{11}$$

For even d, exactly half of the channels are used for scoring. Because each dot-product term is $q_c k_c$, query magnitude is a low-cost proxy for potential channel influence when key-channel scales are comparable; it is not an exact measure of contribution to the logits.

Rescoring the candidate cache. Each selected channel set is shared by the query heads associated with its KV head and applied consistently to queries, historical keys, and current keys. With N_1 candidates remaining after coarse pruning,

$$s^{(2)}_{h,j} = s_{h,j}(\mathcal{I}_f, \{1, \ldots, T\}, \mathcal{C}_h)\,, \qquad 1 \le j \le N_1. \tag{12}$$

Unlike coarse scoring, fine scoring includes all current keys in the normalizer. It retains the original logit scale $1/\sqrt{d}$, rather than rescaling by $1/\sqrt{d'}$. Top-p selection with threshold p_2 gathers entries from the full-dimensional candidate cache. Refinement can remove additional entries, but cannot recover entries discarded at the coarse stage.

2.6 Dense Cache Construction by Head Refill

Head refill converts variable per-head selections into a dense cache of uniform length without padding. Let $\mathcal{A}_h$ denote the selected entries for KV head h, with $b_h = |\mathcal{A}_h|$, and define $B = \max_h b_h$. Each head supplements $\mathcal{A}_h$ with its highest-scoring unselected entries to reach B, and the corresponding keys and values are gathered using identical indices, yielding $\bar{K}, \bar{V} \in \mathbb{R}^{B \times H_{kv} \times d}$.

With the Top-p tie-breaking order, refill preserves each initial selection and completes it to the head's top-B entries. Hence, global Top-p effectively determines a common cache length B, while token identities may differ across heads. The resulting cache contains $H_{kv}B$ head–token entries. Refill is applied after both stages.

3 Experiments

Our experiments test four claims: refinement can further compress a quality-preserving coarse cache at a small marginal quality cost; staged selection can outperform aggressive early pruning at comparable or stricter final budgets; local query sampling preserves high-importance entries despite ranking disagreements; and downstream attention savings can exceed the additional scoring work. We examine these claims through incremental-stage comparisons, retention sweeps, estimator diagnostics, and systems measurements.

3.1 Experimental Setup

Model and datasets. We evaluate BAGEL-7B-MoT, introduced by Deng et al. (2025), on MagicBrush (Zhang et al., 2023a), EMU-Edit (Sheynin et al., 2023), and HumanEdit (Bai et al., 2024). MagicBrush and EMU-Edit use 512-resolution inputs, while HumanEdit uses 1024-resolution inputs. For HumanEdit, we select 100 samples with native resolutions closest to 1024×1024 and resize them to 1024×1024. Model weights, numerical precision, guidance settings, and the denoising schedule are fixed within each comparison.

Compared pruning schedules. **No compression** retains the complete conditioning cache. **Coarse** prunes once at denoising step 0 using locally sampled queries. **C2F** first applies coarse pruning at step 0 and then selects again at step 5. We vary p_1 for Coarse and fix $p_1 = 0.99$ while varying p_2 for C2F. These thresholds specify retained importance mass, not token-retention ratios.

3.2 Evaluation Metrics

Editing quality. We report GT-L1, CLIP-I, CLIP-O, and DINO. GT-L1 measures pixel-level error to the target image, CLIP-I and DINO measure similarity to the source, and CLIP-O measures alignment with the target caption. Lower GT-L1 and higher CLIP/DINO scores are preferred.

Efficiency. We report KV retention, KV storage, attention-and-scoring computation, and end-to-end request latency. Retention is normalized by the uncompressed cache, and systems measurements

Table 1: Incremental benefit of fine refinement. Coarse uses $p_1 = 0.99$; C2F adds $p_2 = 0.95$. Extra cut measures the additional reduction of the coarse cache, $100(1 - r_{\mathrm{C2F}}/r_{\mathrm{Coarse}})\%$. A dash denotes an unavailable or inapplicable value.

Dataset	Method	Retained (%)	Extra cut (%)	GT-L1 ↓	CLIP-I ↑	CLIP-O ↑	DINO ↑
MagicBrush	No compression	100.00	—	0.0728	0.8757	—	0.7517
	Coarse	71.85	—	0.0795	0.8728	—	0.7354
	C2F	56.72	21.06	0.0805	0.8708	—	0.7300
EMU-Edit	No compression	100.00	—	—	0.8359	23.4746	0.7317
	Coarse	71.97	—	—	0.8326	23.5768	0.7236
	C2F	56.67	21.26	—	0.8302	23.6292	0.7182
HumanEdit	No compression	100.00	—	0.0290	0.9320	20.0644	0.8564
	Coarse	65.90	—	0.0291	0.9331	19.7619	0.8561
	C2F	48.81	25.93	0.0341	0.9122	19.5297	0.8050

aggregate all classifier-free guidance branches. We compute $F_{\mathrm{attn+score}} = F_{\mathrm{attn}} + F_{\mathrm{score}}$, where F_{attn} covers $QK^\top$ and attention–value products, and F_{score} covers additional importance-estimation matrix multiplications. One multiply–accumulate is counted as two FLOPs. Request latency includes the complete generation pipeline except model loading and evaluation.

3.3 Main Results and Comparison

Coarse-to-fine pruning achieves substantially higher compression than coarse pruning alone. As shown in Table 1, coarse pruning with $p_1 = 0.99$ retains 71.85% and 71.97% of the KV cache on MagicBrush and EMU-Edit, while adding refinement with $p_2 = 0.95$ further reduces retention to 56.72% and 56.67%. This removes an additional 21.06% and 21.26% of the cache remaining after coarse pruning, increasing the overall KV reduction from approximately 28% to 43%. The result confirms that substantial redundancy remains after conservative early pruning.

The additional compression incurs only a small quality penalty on MagicBrush and EMU-Edit. Compared with Coarse, C2F decreases CLIP-I by only 0.23% and 0.28% and DINO by 0.74% and 0.76%, respectively. MagicBrush GT-L1 changes only from 0.0795 to 0.0805, while EMU-Edit CLIP-O slightly improves from 23.5768 to 23.6292. Thus, more than one fifth of the remaining cache can be removed without comparable quality degradation. This favorable trade-off is consistent with the staged design of C2F, which preserves a broader candidate set during coarse pruning and performs finer selection later using query-guided half-channel importance estimation.

The quality–compression trade-off becomes more pronounced on HumanEdit. On HumanEdit, C2F reduces retention from 65.90% to 48.81%, removing an additional 25.93% of the coarse cache. This stronger compression causes larger quality changes: CLIP-I decreases by 2.24%, DINO by 5.97%, and GT-L1 increases from 0.0291 to 0.0341. This indicates that the same p_2 can yield different quality costs across datasets, and should therefore be treated as a controllable quality–budget parameter rather than a universally near-lossless setting.

3.4 Ablation Study Across Different KV Retention Ratios

C2F achieves better quality under stricter cache budgets. Figures 3–5 compare Coarse and C2F using realized KV retention. On MagicBrush, C2F retains 37.18% of the cache versus 48.11% for Coarse, while improving GT-L1 from 0.1196 to 0.0865, CLIP-I from 0.8344 to 0.8573, and DINO from 0.6442 to 0.7020. On EMU-Edit, C2F similarly reduces retention from 33.35% to 24.07% while improving CLIP-I from 0.7576 to 0.7995, CLIP-O from 23.1072 to 23.5183, and DINO from 0.5526 to 0.6622. These results show that C2F can achieve better editing quality with a substantially smaller final cache.

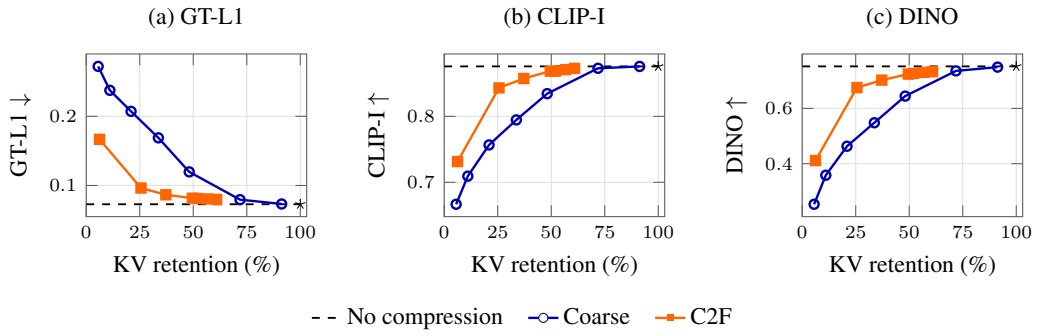

Figure 3: Editing quality versus KV retention on MagicBrush. The curves test whether staged selection can improve quality at a stricter final cache budget. Coarse varies p_1; C2F fixes $p_1 = 0.99$ and varies p_2. Lines connect measured configurations, not additional evaluations. The dashed line shows uncompressed quality at 100% actual retention.

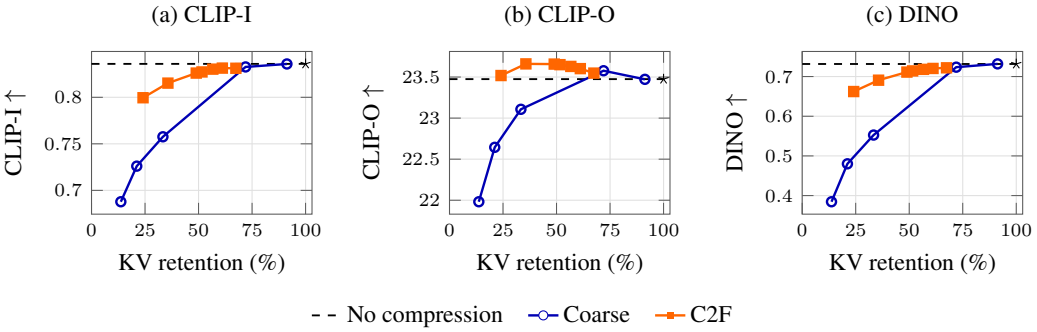

Figure 4: Editing quality versus KV retention on EMU-Edit. C2F has measured operating points with both lower retention and better scores than aggressive coarse pruning. Coarse varies p_1; C2F fixes $p_1 = 0.99$ and varies p_2. Lines connect measured configurations, not additional evaluations. The dashed line shows uncompressed quality at 100% actual retention.

Staged pruning reduces the information loss of aggressive early pruning. Unlike aggressive coarse pruning, which commits to a small cache at the beginning of denoising, C2F preserves a broader candidate set and postpones stronger pruning until a later step. Refinement then reassesses the candidates using updated queries and query-guided feature channels, enabling more selective removal before the final cache budget is imposed.

The same trend extends across a broad retention range on HumanEdit. Although HumanEdit shows a more metric-dependent trade-off, staged pruning still provides a more favorable quality–compression frontier over a broad range. At 17.72% versus 16.07% retention, C2F improves all four metrics; at 27.79% versus 26.23%, it improves GT-L1, CLIP-O, and DINO while CLIP-I changes only marginally. At approximately 40% retention, three similarity metrics also improve with only a small GT-L1 increase. Together with MagicBrush and EMU-Edit, these results show that C2F generally preserves more editing quality than coarse-only pruning under comparable or smaller KV budgets.

3.5 Ablation of Importance Estimation

Sampled-QK provides reliable importance estimates for coarse pruning. Local visual queries exhibit strong redundancy: on MagicBrush, mean cosine similarity is 0.8133 within local groups, compared with 0.7695 between adjacent groups and 0.6957 between non-local groups. To quantify how well sampled scoring approximates full scoring, let S_β and $\widehat{S}_\beta$ denote the same-sized top-β

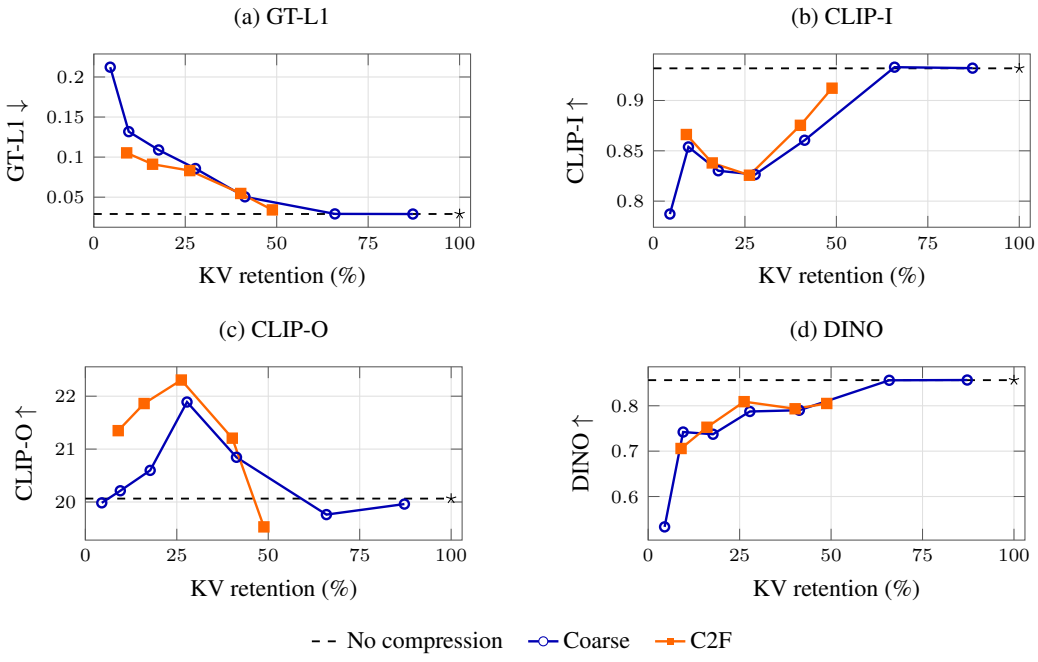

Figure 5: Editing quality versus KV retention on HumanEdit. The four metrics expose both gains and trade-offs at nearby retention levels; they should be interpreted jointly. Coarse varies p_1; C2F fixes $p_1 = 0.99$ and varies p_2. Lines connect measured configurations, not additional evaluations. The dashed line shows uncompressed quality at 100% actual retention.

Table 2: Importance-estimation fidelity on MagicBrush using one query and corresponding current key per 4×4 spatial group. The budget β fixes the selected head–token count and differs from Top-p. Index overlap and relative importance mass compare sampled selection with the same-sized full-score selection; the latter measures the cost of ranking disagreement under the reference scores.

Metric	$\beta = .05$	.10	.30	.50	.80
Index overlap (%)	61.9	70.8	88.0	94.8	97.7
Relative importance mass (%)	87.5	90.5	94.5	99.4	99.97

selections obtained by the full and sampled estimators, respectively. We measure

$$O_\beta = \frac{|S_\beta \cap \widehat{S}_\beta|}{|S_\beta|}, \qquad R_\beta = \frac{\sum_{j \in \widehat{S}_\beta} s_j^{\text{full}}}{\sum_{j \in S_\beta} s_j^{\text{full}}}, \tag{13}$$

where O_β is the selection overlap and R_β is the retained reference importance. At a 10% budget, sampled scoring achieves 70.8% overlap while preserving 90.5% of the reference importance; at a 50% budget, these values increase to 94.8% and 99.4%. Thus, local sampling preserves most high-importance KV information despite moderate ranking differences, supporting its use for efficient coarse screening.

QChan50 substantially reduces fine-stage scoring overhead. We compare QChan50 with FULL refinement after the same coarse-pruning stage. As shown in Table 3, QChan50 reduces fine-stage scoring from 875.42 to 28.16 GMAC on MagicBrush and from 13044.35 to 410.62 GMAC on HumanEdit, corresponding to approximately 96.8%–96.9% less computation. Scoring latency is reduced from 414.96 to 87.19 ms ($4.76\times$) and from 3415.95 to 177.68 ms ($19.23\times$), respectively. This efficiency gain comes with a measurable quality trade-off, particularly on HumanEdit, indicating that QChan50 is an efficiency-oriented approximation rather than a lossless replacement for FULL scoring. Nevertheless, its substantially lower scoring cost makes later-stage cache refinement practical under tighter KV budgets.

Table 3: Fine-stage importance-estimator ablation. Both FULL and QChan50 follow the same coarse-pruning stage and use $p_1 = 0.99, p_2 = 0.90$. QChan50 combines sampled queries with query-guided half-channel scoring. Score cost measures only the fine-stage importance estimation.

Dataset	Fine estimator	Retained (%)	GT-L1 ↓	CLIP-I ↑	CLIP-O ↑	Score (GMAC)	Score (ms)
MagicBrush	FULL	40.70	0.1238	0.8961	24.7478	875.42	414.96
	QChan50	36.82	0.1390	0.8598	25.1698	**28.16**	**87.19**
HumanEdit	FULL	43.61	0.0305	0.9526	22.8082	13044.35	3415.95
	QChan50	39.90	0.0487	0.8615	24.2733	**410.62**	**177.68**

Table 4: KV storage, arithmetic cost, and request latency with FlashAttention. Both pruning schedules use $p_1 = 0.99$; C2F additionally uses $p_2 = 0.95$. Retention and KV storage aggregate all classifier-free guidance branches. $F_{\text{attn+score}}$ counts attention and importance-scoring matrix multiplications, not total model FLOPs. Latency includes pruning and cache construction; negative ΔT indicates faster inference.

Dataset	Method	Retained (%)	KV (MiB)	$F_{\text{attn+score}}$ (TFLOPs)	Latency (s)	ΔT (%)
MagicBrush	No compression	100.00	263.14	159.22	8.422	0.00
	Coarse	67.21	176.87	127.51	8.537	+1.37
	C2F	45.88	120.73	108.96	8.687	+3.15
EMU-Edit	No compression	100.00	263.51	159.35	8.417	0.00
	Coarse	66.82	176.09	127.22	8.553	+1.62
	C2F	45.44	119.74	108.61	8.718	+3.58
HumanEdit	No compression	100.00	985.32	2443.19	30.272	0.00
	Coarse	65.90	649.37	1949.93	29.191	-3.57
	C2F	48.81	480.98	1727.89	28.829	-4.77

3.6 Practical Efficiency

Table 4 evaluates whether the reduced KV cache yields net computational and latency benefits after accounting for importance estimation. Relative to no compression, C2F reduces KV storage by 51.19–54.56% and the counted attention-and-scoring computation by 29.28–31.84%. Compared with Coarse, the refinement stage further removes 25.93–32.00% of the remaining cache and reduces computation by another 11.39–14.63%. Thus, the additional scoring cost of the second stage does not offset the attention savings obtained from the shorter cache.

The wall-clock benefit depends on the workload. On HumanEdit, C2F reduces request latency by 4.77%, from 30.272 to 28.829 seconds. In contrast, latency increases by 3.15% on MagicBrush and 3.58% on EMU-Edit, despite their lower arithmetic cost. This indicates that compression provides consistent storage and computation savings, while end-to-end speedup requires a sufficiently large attention workload to amortize the scoring and cache-management overhead.

4 Conclusion

We propose a query-guided coarse-to-fine KV cache compression framework for unified multimodal generation. The coarse stage uses spatial query sampling to reduce the cost of importance estimation, while the fine stage further compresses the shortened cache with query-guided half-channel scoring. Experiments show favorable quality–efficiency trade-offs across compression levels: C2F removes substantial residual cache redundancy with only a small quality cost on MagicBrush and EMU-Edit, and generally achieves better editing quality than coarse pruning under comparable or smaller KV budgets. Systems measurements further demonstrate substantial reductions in KV storage and attention-and-scoring computation.

AI USE STATEMENT

In this work, we used generative AI tools to assist with method implementation, including modifying and running the research code. Additionally, we used these tools to create preliminary sketches of framework diagrams, revise selected paragraphs, and improve sentence clarity and readability.

AI tools also assisted with consistency checks between figures, tables, and numerical discussions, and with LaTeX typesetting adjustments. No experimental measurements were generated or altered by AI tools during these manuscript checks.

We have reviewed all AI-assisted work. Specifically, we checked the code modifications against the intended method, examined the outputs of the AI-assisted code execution, verified that the diagram sketches accurately represented our method, and reviewed the revised text for technical accuracy and consistency with the reported work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Jinbin Bai, Wei Chow, Ling Yang, Xiangtai Li, Juncheng Li, Hanwang Zhang, and Shuicheng Yan. Humanedit: A high-quality human-rewarded dataset for instruction-based image editing. *arXiv preprint arXiv:2412.04280*, 2024.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S. Kevin Zhou. Ada-KV: Optimizing KV cache eviction by adaptive budget allocation for efficient LLM inference. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2407.11550.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, volume 2025, pp. 34668–34684, 2025.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean Flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.13447.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

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.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai, Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, Yibo Zhu, Binxing Jiao, Xiangyu Zhang, Gang Yu, and Daxin Jiang. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu Edit: Precise image editing via recognition and generation tasks. *arXiv preprint arXiv:2311.10089*, 2023. URL https://arxiv.org/abs/2311.10089.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. Vl-cache: Sparsity and modality-aware kv cache compression for vision-language model inference acceleration. In *International Conference on Learning Representations*, volume 2025, pp. 219–239, 2025.

Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. Look-m: Look-once optimization in kv cache for efficient multimodal long-context inference. *arXiv preprint arXiv:2406.18139*, 2024.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025a. URL `https://arxiv.org/abs/2508.02324`.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. OmniGen2: Towards instruction-aligned multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025b.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. *arXiv preprint arXiv:2506.15564*, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024b.

Bowen Zeng, Feiyang Ren, Jun Zhang, Xiaoling Gu, Ke Chen, Lidan Shou, and Huan Li. Hybridkv: Hybrid kv cache compression for efficient multimodal large language model inference. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13018–13034, 2026.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *Advances in Neural Information Processing Systems*, 2023a.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in neural information processing systems*, 36:34661–34710, 2023b.

Shanshan Zhao, Xinjie Zhang, Jintao Guo, Jiakui Hu, Lunhao Duan, Minghao Fu, Yong Xien Chng, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, et al. Unified multimodal understanding and generation models: Advances, challenges, and opportunities. *arXiv preprint arXiv:2505.02567*, 2025.

Xiabin Zhou, Wenbin Wang, Minyan Zeng, Jiaxian Guo, Xuebo Liu, Li Shen, Min Zhang, and Liang Ding. Dynamickv: Task-aware adaptive kv cache compression for long context llms. In *EMNLP (Findings)*, pp. 8042–8057, 2025.

A Appendix

You may include other additional sections here.