RECOVERING EVICTED RESPONSES IN PROGRESSIVE SPARSE ATTENTION FOR IMAGE SUPER-RESOLUTION

Anonymous authors
Paper under double-blind review

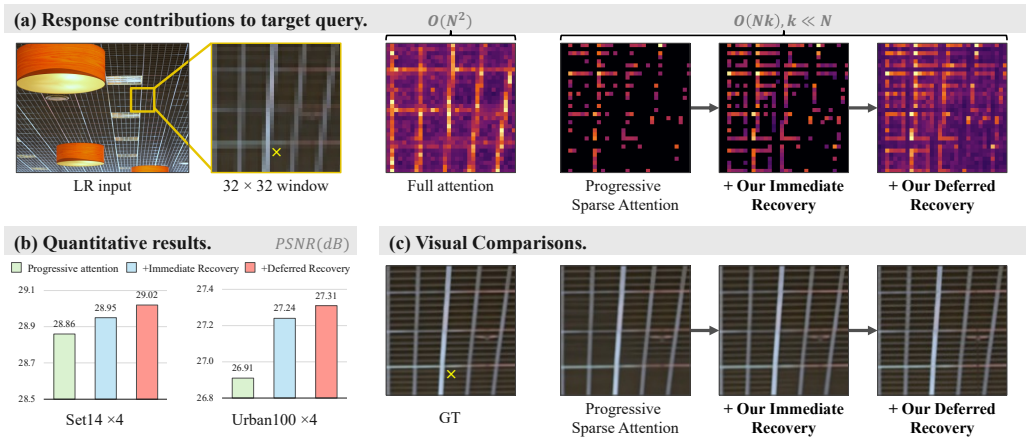


Figure 1: **Recovering evicted response in progressive sparse attention.** (a) Attention response within a 32×32 window. Progressive sparse attention retains only part of the full attention response. Immediate Recovery (IR) compensates for discarded response at the current stage, while Deferred Recovery (DR) makes it available to deeper layers. (b) Cumulatively incorporating IR and DR yields consistent PSNR improvements across datasets. (c) Corresponding visual comparisons demonstrate progressively sharper structures as more evicted response is recovered.

ABSTRACT

Attention-based methods have achieved remarkable performance in image super-resolution by exploiting long-range spatial dependencies, yet dense interaction over the full token set remains prohibitively expensive. To mitigate this issue, progressive sparse attention methods have recently emerged, gradually pruning the candidate tokens available to each query as the network deepens, thereby substantially reducing the computational cost. However, such early pruning decisions permanently exclude interactions that may already carry useful information or become relevant in later layers. We refer to this loss of potentially useful response information as *evidence eviction*. To address this, we propose **EvRC (Evicted Response Compensation)**, which recovers pruned response information through two complementary mechanisms at different timescales. Instantly, **Immediate Recovery** folds the discarded response directly into the current-stage representation, ensuring that no scored interaction is silently dropped. Subsequently, **Deferred Recovery** compresses evicted responses into compact traces deposited in a shared Trace Bank, enabling subsequent layers to selectively retrieve cross-layer evidence conditioned on their current representation. Together, the two components ensure that while the active support shrinks progressively, response evidence from the broader scored support remains accessible throughout the network. Experiments on standard benchmarks under both classical and lightweight settings demonstrate state-of-the-art performance, with particularly notable gains in recovering fine structures within complex scenes.

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.

1 INTRODUCTION

Single-image super-resolution (SISR), which aims to reconstruct missing high-frequency content from a single low-resolution observation, often relies on integrating cues from distant image regions rather than the local neighborhood alone. Self-attention provides a natural mechanism for such non-local aggregation and has become a cornerstone of modern SR architectures (Liang et al., 2021; Chen et al., 2023a; Zhang et al., 2022). However, this capability comes at a quadratic computational cost with respect to the number of tokens, and striking a favorable balance between receptive field and efficiency therefore remains a central design challenge in efficient SR.

Recently, progressive sparse attention has emerged as a promising approach to this problem (Long et al., 2025; Rao et al., 2021). Starting from a broad candidate set, the method progressively prunes it across depth: at each stage, every query scores its currently available candidates, retains only the top-ranked ones, and passes the survivors to the next stage, yielding a nested sequence of shrinking supports $\mathcal{S}^0 \supseteq \mathcal{S}^1 \supseteq \cdots \supseteq \mathcal{S}^R$, where $\mathcal{S}^r$ denotes the set of candidates surviving the r-th pruning stage. While this cascade reduces computation monotonically with depth, it also renders exclusions permanent: once pruned, a candidate can never be reconsidered by later layers.

However, a discarded candidate is not necessarily uninformative. First, relevance is not static. Representations evolve with depth, *i.e.*, two positions that appear weakly related in an early layer may become strongly coupled in a deeper one, yet the exclusion is decided before that relevance can emerge. Second, selection at each pruning stage typically operates under a fixed budget, so even when the ranking faithfully reflects current relevance, an informative candidate may fall outside the top-k cutoff and be irrevocably discarded. We refer to the irreversible loss of informative interactions arising from either mechanism as **evidence eviction**.

Evidence eviction is particularly consequential in SR, where recovering fine structure often depends on a small number of spatially specific non-local cues, and the surviving candidates rarely provide equivalent substitutes once a structurally relevant interaction is removed. Prior work has approached this problem from the selection side, improving sparse interaction sets through content-aware candidate construction, inherited score refinement, or query-specific match exploration (Mei et al., 2021; Su et al., 2024; Long et al., 2025). While such designs reduce evidence eviction by retaining more informative candidates, they do not recover the response information carried by candidates that are ultimately pruned away—a gap that becomes more pronounced as pruning grows more aggressive.

In this paper, we take a complementary route: rather than improving the selection rule, we preserve the response information that pruning would otherwise discard — immediately, when a fixed budget removes an informative candidate, and in later layers, when relevance emerges only after pruning has already taken place. We instantiate this principle with **EvRC (Evicted Response Compensation)**, a lightweight plug-in that augments progressive sparse attention without reactivating discarded candidates or altering the pruning schedule. EvRC consists of two complementary components. **Immediate Recovery** (IR) acts at each pruning stage, folding the attention-weighted response of the evicted candidates directly into the current representation. **Deferred Recovery** (DR) preserves evicted information across layers by writing a compact summary of each pruned band to a query-indexed Trace Bank, which records the fraction of attention mass carried by the discarded band and a low-dimensional residual between the responses of the evicted and retained candidates. Later queries retrieve relevant response traces conditioned on their current content, keeping evidence accessible after the original candidates lose eligibility. Together, the two components address both the immediate and deferred costs of evidence eviction, with only modest additional computation ($\sim 3.5\%$ MACs for classical SR) relative to the backbone. Our contributions are as follows:

- **Observation.** We identify that progressive sparse attention suffers from evidence eviction: nested pruning may irreversibly discard candidates that remain informative to deeper layers, thereby compromising SR reconstruction quality.
- **Method.** We propose EvRC, a lightweight plug-in module that rescues evicted evidence via two complementary components: Immediate Recovery, which folds evicted responses into the current representation, and Deferred Recovery, which retains a compact trace for content-conditioned retrieval in subsequent layers.
- **Validation.** We integrate EvRC into a progressive sparse attention backbone for SR. Controlled ablations confirm the existence and reconstruction impact of evidence eviction, and

comprehensive evaluations on standard benchmarks demonstrate state-of-the-art performance under both classical and lightweight SR settings.

2 Related Work

Image Super-Resolution. CNN-based methods established the SR pipeline through residual learning and channel recalibration (Dong et al., 2014; Kim et al., 2016; Zhang et al., 2018a; Tong et al., 2017; Zhang et al., 2018b). Transformer-based methods subsequently introduced content-dependent spatial aggregation through shifted-window attention and cross-window interaction (Chen et al., 2021; 2022; Wang et al., 2023b; Li et al., 2021). Their considerable cost on high-resolution feature maps has motivated efficient designs to either restructure the attention operator (Zhang et al., 2022; Lee et al., 2025) or compress the key–value context (Tran et al., 2026); a complementary route restricts which interactions are evaluated at all, which is the direction pursued here.

Token Pruning in Self-Attention. Token pruning reduces attention cost by restricting the candidates available to subsequent layers (Rao et al., 2021; Yin et al., 2022). In SR, this has been explored at multiple granularities: window routing (Wang et al., 2024), depth-wise halting (Luo et al., 2024), and hash- or cluster-based grouping (Mei et al., 2021; Zhao et al., 2024). PFT (Long et al., 2025) operates at the finest granularity by maintaining a query-specific candidate set that is progressively contracted across layers, achieving strong trade-offs between efficiency and quality. However, its nested support renders evidence eviction as discussed before, which motivates the present work.

Information Preservation under Pruning. EViT (Liang et al., 2022) merges inattentive tokens into a representative, Liu et al. (2024) retain pruned tokens for possible reactivation, and Hi-Lo Prune (Sun et al., 2026) transfers information to retained tokens before removal. These methods treat the token as the unit of removal and apply uniformly across all queries, whereas progressive sparse attention removes query-specific interactions: the same key may remain available to one query but not another. EvRC therefore preserves compact, query-indexed response traces rather than the discarded tokens themselves, compensating for lost information without storing keys and values or reactivating pruned interactions.

3 Motivation

Preliminaries. Let $q_i, k_j, v_j \in \mathbb{R}^d$ denote queries, keys, and values for an input window of size N. While full attention scales quadratically as $\mathcal{O}(N^2 d)$, progressive sparse attention reduces computational cost by gradually narrowing the set of candidate keys, using the survivors from each stage as the search space for the next. Concretely, for a given query i, each stage r receives $\mathcal{S}^{r-1}$ from the preceding stage and outputs a pruned set $\mathcal{S}^r$, with $\mathcal{S}^0 = \{1, \ldots, N\}$. The progressive sparse attention mechanism updates the candidate set via a $\mathrm{TopK}(\cdot)$ selection process:

$$\mathcal{S}^r = \mathrm{TopK}\big(\ell_j : j \in \mathcal{S}^{r-1}\big). \tag{1}$$

Here, ℓ_j denotes the relevance score of candidate j at stage r, which is typically the attention weight itself (Long et al., 2025) or the output of a learned selector (Rao et al., 2021). In our implementation, $\ell_j = a_j$, where a_j is the current-layer attention weight computed over the incoming support:

$$a_j = \mathrm{Softmax}_{j \in \mathcal{S}^{r-1}}\left(\frac{\langle q_i, k_j \rangle}{\sqrt{d}} + b_j\right), \qquad y^r_{\mathcal{S}} = \sum_{j \in \mathcal{S}^r} a_j \, v_j. \tag{2}$$

Here b_j collects the positional bias and masking terms, and $y^r_{\mathcal{S}}$ denotes the sparse attention output. Because each stage selects only from earlier survivors, the supports are nested, i.e., $\mathcal{S}^0 \supseteq \mathcal{S}^1 \supseteq \cdots \supseteq \mathcal{S}^R$. By fixing $|\mathcal{S}^r| = K_r$ at each stage, the attention cost at every subsequent layer is reduced from $\mathcal{O}(N^2 d)$ to $\mathcal{O}(N K_r d)$.

Motivation. However, the nested-support constraint also renders these exclusions permanent. To make this cost explicit, we partition the incoming response into a retained and a discarded band:

$$y^r_{\mathrm{in}} = \underbrace{\sum_{j \in \mathcal{S}^r} a_j v_j}_{y^r_{\mathcal{S}}} + \underbrace{\sum_{j \in \mathcal{D}^r} a_j v_j}_{y^r_{\mathcal{D}}}, \qquad \mathcal{D}^r = \left\{ j \in \mathcal{S}^{r-1} \mid j \notin \mathcal{S}^r \right\}, \tag{3}$$

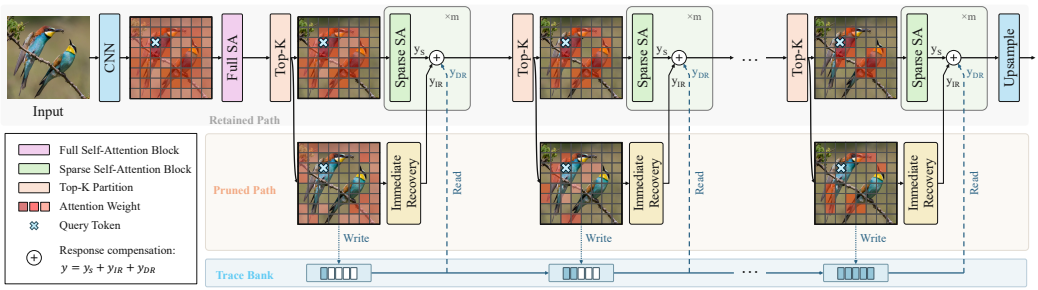


Figure 2: The overall architecture of EvRC.

where $\mathcal{D}^r$ denotes the subset of candidates evicted at pruning stage r. The discarded band incurs loss at two timescales: *immediately*, as $y^r_{\mathcal{D}}$ is excluded from the current-stage representation, and *permanently*, as $\mathcal{D}^r$ remains inaccessible to all subsequent layers. This observation motivates our EvRC, which reincorporates the discarded response at current stage to recover immediate loss, and propagates a compact trace forward to provide deeper layers with access to the evicted evidence.

4 EvRC: Evicted Response Compensation

As shown in Figure 2, EvRC employs two complementary mechanisms to recover information lost across two distinct timescales. To mitigate permanent information loss, we introduce Deferred Recovery, a mechanism that aggregates evicted evidence across layers. Specifically, at each pruning stage, a Trace Writer compresses discarded responses into compact traces and deposits them into the Trace Bank. Subsequently, at deeper layers, a Trace Reader retrieves these stored traces, conditioned on the current features, providing aggregated access to previously evicted evidence. The immediate information loss is mitigated by Immediate Recovery, which establishes a stage-local pathway to incorporate the discarded responses directly into the current attention output.

4.1 Deferred Recovery

Trace Writing. As described above, Deferred Recovery represents each discarded band by a compact trace. A natural choice is to directly compress the discarded response $y^r_{\mathcal{D}}$. However, this conflates two distinct signals: how much attention mass was discarded and what content that mass carried, making it harder for the reader to calibrate retrieval strength against the magnitude of the loss. Instead, as shown in Figure 3, we decouple the two by recording the discarded mass ρ^r separately from a content residual z^r that captures only the information absent from the retained band. Concretely, for stage r, the retained mass and discarded masses, $\rho_{\mathcal{S}}$ and $\rho_{\mathcal{D}}$, are defined as:

$$\rho_{\mathcal{S}} = \sum_{j \in \mathcal{S}^r} a_j, \qquad \rho_{\mathcal{D}} = \sum_{j \in \mathcal{D}^r} a_j = 1 - \rho_{\mathcal{S}}, \tag{4}$$

and the content residual c is defined as the difference between the normalized discarded and retained responses:

$$c = \mu_{\mathcal{D}} - \mu_{\mathcal{S}}, \tag{5}$$

where $\mu_{\mathcal{S}} = y^r_{\mathcal{S}}/\rho_{\mathcal{S}}$ and $\mu_{\mathcal{D}} = y^r_{\mathcal{D}}/\rho_{\mathcal{D}}$. Storing the difference rather than $\mu_{\mathcal{D}}$ encourages c to encode only the information absent from the retained band, making the compressed trace more compact and easier to retrieve selectively. Together, $\rho_{\mathcal{D}}$ and c recover the full incoming response via the identity:

$$y^r_{\text{in}} = \rho_{\mathcal{S}}\,\mu_{\mathcal{S}} + \rho_{\mathcal{D}}\,\mu_{\mathcal{D}} = \mu_{\mathcal{S}} + \rho_{\mathcal{D}}\,c, \tag{6}$$

confirming that the two quantities carry complementary information about the discarded band. We then apply a linear layer $E(\cdot) : R^d \to R^{d_m}$ to compress the content residual as:

$$z^r = E(c) \in \mathbb{R}^{d_m}, \qquad d_m \ll d, \tag{7}$$

and the pair (ρ^r, z^r) is appended to the Trace Bank, where ρ^r denotes $\rho_{\mathcal{D}}$ at stage r. Writing is applied independently for every query, with stage- and head-specific projections and traces.

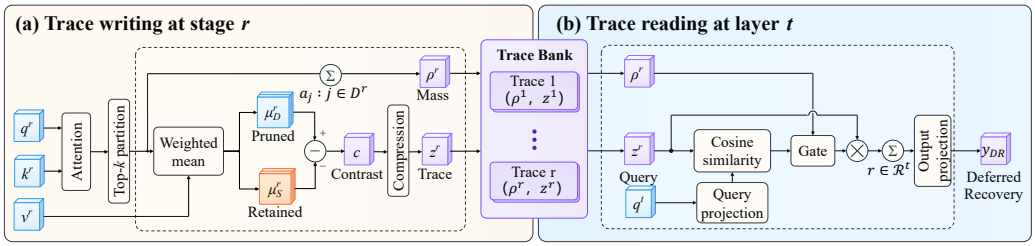


Figure 3: Illustrations of the Trace Writing and Trace Reading operators.

Trace Reading. At a reading layer t, the Trace Reader conditions retrieval on the current query features to determine which earlier traces remain relevant. The relevance of a stored trace from source head h at stage r to the current head u is measured by cosine similarity:

$$s^r_{u\leftarrow h} = \mathrm{CosSim}(P(q_u), z^r_h)\,. \tag{8}$$

where q_u denotes the query at head u, $P(\cdot) : R^d \to R^{d_m}$ denotes a linear projection, $s^r_{u\leftarrow h}$ denotes the relevance of the stored trace to the current query. Since ρ^r_h records the fraction of attention mass discarded when that trace was written, the final routing weight is calculated via a query-dependent gate:

$$w^r_{u\leftarrow h} = G(\rho^r_h, s^r_{u\leftarrow h}), \tag{9}$$

where $G(\cdot)$ combines query–trace relevance with the stored discarded-mass prior to determine the routing weight. Letting $\mathcal{R}^t$ denote the pruning stages whose traces are available for layer t, the retrieved traces are aggregated within each source head and mixed across heads to produce the output correction:

$$y_{DR} = \beta^t_u\, D\left(\mathrm{Concat}^H_{h=1} \sum_{r\in\mathcal{R}^t} w^r_{u\leftarrow h}\, z^r_h\right), \tag{10}$$

where H is the number of heads and $D(\cdot) : R^{Hd_m} \to R^d$ is the output projection. Source heads are kept separate until the final projection, allowing each target head to draw selectively on different sources. The scalar gate β^t_u is initialized to zero, so Deferred Recovery contributes no correction initially and is activated only when beneficial.

Deferred Recovery provides a learned response correction rather than reconstructing or reactivating removed interactions. Unlike response folded into the ordinary feature stream, stage-specific traces remain explicitly addressable by later queries, preserving the ability to recover information selectively. The Trace Bank is reset at every forward pass and remains differentiable throughout training. More details are provided in Appendix A.

4.2 Immediate Recovery

Immediate Recovery addresses the immediate loss at the pruning stage by making direct use of the discarded band before it is evicted. At stage r, its output can be formulated as:

$$y_{\mathrm{IR}} = \alpha\, y_{\mathcal{D}}, \tag{11}$$

where α is a learned scalar specific to each pruning stage and head. We initialize α to one, such that the combined response $y_{\mathcal{S}} + y_{\mathrm{IR}}$ equals the full incoming response y_{in} at initialization. During training, α is optimized to adapt the contribution of the discarded band across heads and depth.

Immediate Recovery is particularly efficient: it requires no additional query–key scoring and only one discarded-band value aggregation when used alone. Moreover, in the complete EvRC network, this aggregation is shared with Deferred Recovery. Concretely, selecting $\mathcal{S}^r$ already requires the backbone to evaluate query–key scores for every candidate in the incoming support $\mathcal{S}^{r-1}$, including those subsequently assigned to $\mathcal{D}^r$. Immediate Recovery reuses these existing attention weights to aggregate $\mathcal{D}^r$ and combine its response with $y^r_{\mathcal{S}}$. When Deferred Recovery is also enabled, $y^r_{\mathcal{D}}$ is already computed by the Trace Writer and can be reused directly. In either case, later layers continue to operate on the contracted support $\mathcal{S}^r$. Immediate Recovery is applied only at pruning stages, leaving the original attention computation and feature-processing pipeline unchanged.

4.3 THE OVERALL NETWORK ARCHITECTURE

We instantiate EvRC on a minimal progressive sparse Transformer, using current-layer attention weights directly for top-K selection and response aggregation, without auxiliary selectors or cross-layer score modulation. This minimal design isolates response recovery from candidate ranking, ensuring that the observed gains arise from preserving evicted responses rather than modifying the pruning decisions. We insert our recovery mechanisms only at pruning stages and designated reading layers while leaving the remaining backbone unchanged. As shown in Figure 2, the retained sparse response forms the main path, and the two recovery terms are combined at the attention-head output. When both mechanisms are active, the resulting response can be formulated as:

$$y_{\mathrm{EvRC}} = y_{\mathcal{S}} + y_{\mathrm{IR}} + y_{\mathrm{DR}} \tag{12}$$

Trace Reading uses only entries committed by earlier pruning stages; therefore, when a layer both reads and writes, the existing bank is read first and the new trace is appended afterward.

Our default model uses six residual Transformer groups with depths $(4, 4, 4, 6, 6, 6)$ and alternates regular and shifted-window attention following Liu et al. (2021). The feature dimension is 240, with $H = 6$ attention heads of dimension $d = 40$. Following Long et al. (2025), attention is computed in 32×32 windows, and each shift lineage follows the support schedule $(1024, 256, 128, 64, 32, 16)$. Every pruning stage writes a response trace to the Trace Bank with $d_m = 8$, and all available traces are read at subsequent reading layers. We also provide a lightweight variant, EvRC-lt, using five groups with depths $(2, 4, 6, 6, 6)$, $H = 4$ attention heads, and a feature dimension of 52. Its support schedule is $(1024, 256, 128, 64, 32)$. Both variants use a trace dimension of $d_m = 8$. Further architectural details are provided in Appendix A.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Following recent works (Guo et al., 2025; Chen et al., 2023a), we train our classical models on DF2K dataset and lightweight models on DIV2K dataset (Timofte et al., 2017; Lim et al., 2017). For evaluation, we adopt five standard benchmarks: Set5 (Bevilacqua et al., 2012), Set14 (Zeyde et al., 2010), B100 (Martin et al., 2001), Urban100 (Huang et al., 2015), and Manga109 (Matsui et al., 2017). We follow the bicubic degradation protocol (Zhang et al., 2017) and report PSNR and SSIM on the Y channel in YCbCr space. Our model is trained by Adam optimizer (Kingma & Ba, 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for 1000K iterations. The initial learning rate is set as 2×10^{-4}, and we decrease learning rate to half after every 200k iterations. In each iteration, we stack 32 LR patches in a batch with size 64×64 as inputs. Our proposed model is implemented in PyTorch (Paszke et al., 2017) framework and trained on two H20 GPUs with 96GB RAM.

5.2 COMPARISON WITH STATE OF THE ART

We evaluate EvRC against state-of-the-art SR methods under both classical and lightweight SR settings. Quantitative results are reported in Table 1 and Table 2, respectively.

Classical SR. As shown in Table 1, EvRC consistently achieves the best performance across all five benchmarks and all three scales ($\times 2$, $\times 3$, $\times 4$), outperforming the compared CNN-based methods (EDSR (Lim et al., 2017) and RCAN (Zhang et al., 2018a)), Transformer-based methods (HAT (Chen et al., 2023a), DAT (Chen et al., 2023b), ATD (Zhang et al., 2024), and PFT (Long et al., 2025)), the graph-based IPG (Tian et al., 2024), and the Mamba-based MambaIRv2 (Guo et al., 2025). Specifically, at $\times 3$ and $\times 4$ scales, EvRC surpasses the second-best method PFT by **+0.21 dB** and **+0.15 dB** on Urban100, which contains complex urban structures with rich high-frequency details, demonstrating EvRC's superior ability to recover fine-grained textures. Notably, the gains are achieved with only 0.1–0.2M additional parameters and 3.5–4.0% more MACs relative to PFT.

Lightweight SR. Table 2 presents results compared with representative lightweight SR methods (*e.g.*, CARN (Ahn et al., 2018), LatticeNet (Luo et al., 2020), OmniSR (Wang et al., 2023a)), where EvRC-lt also achieves state-of-the-art performance across all benchmarks and scales. On the Manga109 dataset, EvRC-lt achieves notable gains of **+0.16 dB**, **+0.27 dB**, and **+0.19 dB** com-

Table 1: Quantitative comparison with state-of-the-art methods for **classical SR**. MACs are computed for a 1280 × 720 output. The best and second-best results are coloured red and blue, respectively.

Method	Scale	#Param	MACs	Set5		Set14		B100		Urban100		Manga109	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR	×2	42.6M	22.14T	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	39.10	0.9773
RCAN	×2	15.4M	7.02T	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.35	0.9385	39.56	0.9785
IPT	×2	115M	7.38T	38.37	-	34.43	-	32.48	-	33.76	-	-	-
SwinIR	×2	11.8M	3.04T	38.42	0.9623	34.46	0.9250	32.53	0.9041	33.81	0.9433	39.92	0.9797
DAT	×2	14.7M	3.73T	38.58	0.9629	34.81	0.9272	32.61	0.9051	34.37	0.9458	40.33	0.9807
HAT	×2	20.6M	5.81T	38.63	0.9630	34.86	0.9274	32.62	0.9053	34.45	0.9466	40.26	0.9809
IPG	×2	18.1M	5.35T	38.61	0.9632	34.73	0.9270	32.60	0.9052	34.48	0.9464	40.24	0.9810
MambaIRv2	×2	22.9M	6.08T	38.65	0.9631	34.89	0.9275	32.62	0.9053	34.49	0.9468	40.42	0.9810
ATD	×2	20.1M	6.47T	38.61	0.9629	34.95	0.9276	32.65	0.9056	34.70	0.9476	40.37	0.9810
PFT	×2	19.6M	5.49T	38.68	0.9635	35.00	0.9280	32.67	0.9058	34.90	0.9490	40.49	0.9815
EvRC (Ours)	×2	19.8M	5.71T	38.72	0.9636	35.09	0.9283	32.70	0.9063	34.96	0.9497	40.57	0.9816
EDSR	×3	43.0M	9.82T	34.65	0.9280	30.52	0.8462	29.25	0.8093	28.80	0.8653	34.17	0.9476
RCAN	×3	15.6M	3.12T	34.74	0.9299	30.65	0.8482	29.32	0.8111	29.09	0.8702	34.44	0.9499
IPT	×3	115M	3.28T	34.81	-	30.85	-	29.38	-	29.49	-	-	-
SwinIR	×3	11.9M	1.35T	34.97	0.9318	30.93	0.8534	29.46	0.8145	29.75	0.8826	35.12	0.9537
DAT	×3	14.8M	1.69T	35.16	0.9331	31.11	0.8550	29.55	0.8169	30.18	0.8886	35.59	0.9554
HAT	×3	20.8M	2.58T	35.07	0.9329	31.08	0.8555	29.54	0.8167	30.23	0.8896	35.53	0.9552
IPG	×3	18.3M	2.39T	35.10	0.9332	31.10	0.8554	29.53	0.8168	30.36	0.8901	35.53	0.9554
MambaIRv2	×3	23.1M	2.70T	35.18	0.9334	31.12	0.8557	29.55	0.8169	30.28	0.8905	35.61	0.9556
ATD	×3	20.3M	2.87T	35.11	0.9330	31.13	0.8556	29.57	0.8176	30.46	0.8917	35.63	0.9558
PFT	×3	19.8M	2.59T	35.15	0.9333	31.16	0.8561	29.58	0.8178	30.56	0.8931	35.67	0.9560
EvRC (Ours)	×3	19.9M	2.69T	35.23	0.9338	31.19	0.8562	29.60	0.8182	30.77	0.8960	35.76	0.9564
EDSR	×4	43.0M	5.54T	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033	31.02	0.9148
RCAN	×4	15.6M	1.76T	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087	31.22	0.9173
IPT	×4	116M	1.85T	32.64	-	29.01	-	27.82	-	27.26	-	-	-
SwinIR	×4	11.9M	0.76T	32.92	0.9044	29.09	0.7950	27.92	0.7489	27.45	0.8254	32.03	0.9260
DAT	×4	14.8M	0.97T	33.08	0.9055	29.23	0.7973	28.00	0.7515	27.87	0.8343	32.51	0.9291
HAT	×4	20.8M	1.45T	33.04	0.9056	29.23	0.7973	28.00	0.7517	27.97	0.8368	32.48	0.9292
IPG	×4	18.3M	1.30T	33.15	0.9062	29.24	0.7973	27.99	0.7519	28.13	0.8392	32.53	0.9300
MambaIRv2	×4	23.1M	1.62T	33.14	0.9057	29.23	0.7975	28.00	0.7511	27.89	0.8344	32.57	0.9295
ATD	×4	20.3M	1.73T	33.10	0.9058	29.24	0.7974	28.01	0.7526	28.17	0.8404	32.62	0.9306
PFT	×4	19.8M	1.41T	33.15	0.9065	29.29	0.7978	28.02	0.7527	28.20	0.8412	32.63	0.9306
EvRC (Ours)	×4	19.9M	1.46T	33.21	0.9071	29.34	0.7993	28.03	0.7531	28.35	0.8453	32.75	0.9313

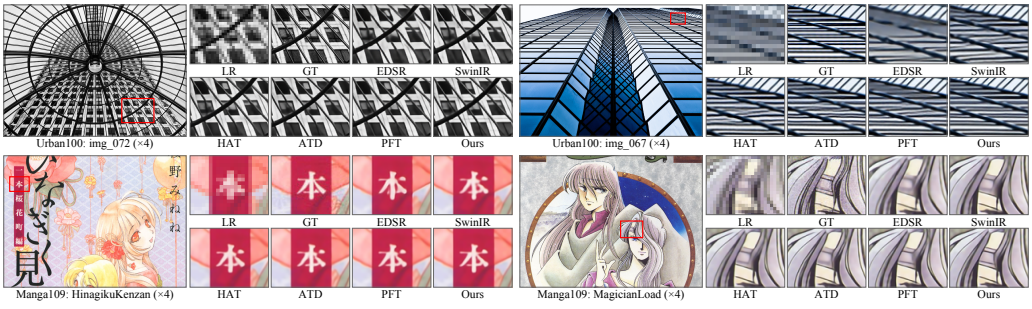


Figure 4: Qualitative comparison of SR reconstruction results.

pared with PFT-lt at ×2, ×3, and ×4 scales, respectively, further validating the effectiveness of the proposed design under resource-constrained settings.

Visual Comparison. Figure 4 provides qualitative comparisons among EvRC and representative methods, including EDSR, SwinIR, HAT, ATD, and PFT. As shown in the figure, on Urban100, EvRC better preserves thin grid lines and their intersections, which are easily smoothed in regions with dense, repetitive patterns. On Manga109, it renders small character strokes and curved hair contours more faithfully to the ground truth. These comparisons demonstrate that preserving evicted response evidence enables EvRC to recover fine structures, with gains that are most pronounced in regions where non-local cues are essential for faithful reconstruction.

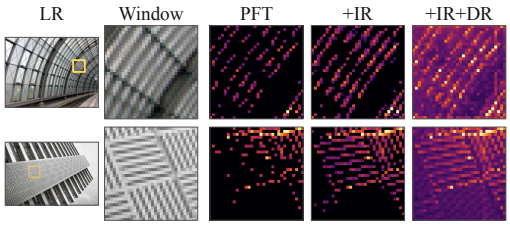
Table 2: Quantitative comparison with state-of-the-art methods for **lightweight SR**. MACs are computed for 1280 × 720 HR outputs. The best and second-best results are coloured red and blue, respectively. "-lt" means light version of the model.

Method	Scale	#Param	MACs	Set5		Set14		B100		Urban100		Manga109	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CARN	×2	1592K	222.8G	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256	38.36	0.9765
LatticeNet	×2	756K	169.5G	38.13	0.9610	33.78	0.9193	32.25	0.9005	32.43	0.9302	-	-
SwinIR-lt	×2	910K	244.2G	38.14	0.9611	33.86	0.9206	32.31	0.9012	32.76	0.9340	39.12	0.9783
ELAN	×2	621K	203.1G	38.17	0.9611	33.94	0.9207	32.30	0.9012	32.76	0.9340	39.11	0.9782
OmniSR	×2	772K	194.5G	38.22	0.9613	33.98	0.9210	32.36	0.9020	33.05	0.9363	39.28	0.9784
ATD-lt	×2	753K	337.1G	38.28	0.9616	34.11	0.9217	32.39	0.9023	33.27	0.9376	39.51	0.9789
PFT-lt	×2	776K	251.3G	38.36	0.9620	34.19	0.9232	32.43	0.9030	33.67	0.9411	39.55	0.9792
EvRC-lt (Ours)	×2	797K	287.2G	38.41	0.9622	34.31	0.9239	32.49	0.9036	33.80	0.9423	39.71	0.9793
CARN	×3	1592K	118.8G	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493	33.50	0.9440
LatticeNet	×3	765K	76.3G	34.53	0.9281	30.39	0.8424	29.15	0.8059	28.33	0.8538	-	-
SwinIR-lt	×3	918K	111.2G	34.62	0.9289	30.54	0.8463	29.20	0.8082	28.66	0.8624	33.98	0.9478
ELAN	×3	629K	90.1G	34.61	0.9288	30.55	0.8463	29.21	0.8081	28.69	0.8624	34.00	0.9478
OmniSR	×3	780K	88.4G	34.70	0.9294	30.57	0.8469	29.28	0.8094	28.84	0.8656	34.22	0.9487
ATD-lt	×3	753K	159.9G	34.70	0.9300	30.68	0.8485	29.32	0.8109	29.16	0.8710	34.60	0.9505
PFT-lt	×3	783K	123.9G	34.81	0.9305	30.75	0.8493	29.33	0.8116	29.43	0.8759	34.60	0.9510
EvRC-lt (Ours)	×3	804K	141.5G	34.89	0.9312	30.81	0.8507	29.38	0.8129	29.62	0.8791	34.87	0.9520
CARN	×4	1592K	90.9G	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837	30.47	0.9084
LatticeNet	×4	777K	43.6G	32.30	0.8962	28.68	0.7830	27.62	0.7367	26.25	0.7873	-	-
SwinIR-lt	×4	930K	63.6G	32.44	0.8976	28.77	0.7858	27.69	0.7406	26.47	0.7980	30.92	0.9151
ELAN	×4	640K	54.1G	32.43	0.8975	28.78	0.7858	27.69	0.7406	26.54	0.7982	30.92	0.9150
OmniSR	×4	792K	50.9G	32.49	0.8988	28.78	0.7859	27.71	0.7415	26.65	0.8018	31.02	0.9151
ATD-lt	×4	769K	85.1G	32.62	0.8997	28.87	0.7884	27.77	0.7439	26.97	0.8107	31.47	0.9198
PFT-lt	×4	792K	63.7G	32.63	0.9005	28.92	0.7891	27.79	0.7445	27.20	0.8171	31.51	0.9204
EvRC-lt (Ours)	×4	813K	72.8G	32.63	0.9007	29.02	0.7906	27.83	0.7455	27.31	0.8201	31.70	0.9222

Selection rule	Set14		Urban100	
	PSNR	SSIM	PSNR	SSIM
Progressive top-K	28.95	0.7892	27.19	0.8171
Full-attention top-K†	29.03	0.7909	27.28	0.8195
Full-attention random-K†	28.78	0.7858	26.66	0.8019
Full-attention bottom-K†	28.61	0.7816	26.19	0.7858

† Recomputes full attention at each layer; used for analysis only.

(a) Effect of candidate-selection rules.



(b) Response-recovery visualization.

Figure 5: **Analysis of evidence eviction and recovery.** (a) Reconstruction performance under different candidate-selection rules. (b) Cumulative response-contribution maps for two examples, showing the progressive effects of Immediate and Deferred Recovery.

5.3 Analysis

We analyze each design choice through controlled ablations. Unless otherwise specified, all experiments are conducted on lightweight ×4 SR and share the same training configuration.

Existence and Impact of Evicted Evidence. We first examine whether inherited support excludes candidates that remain useful to deeper layers. Starting from PFT-lt at ×4, we compare its progressive top-K selection with three variants that recompute attention over the complete window at every layer and select the top-K, random-K, or bottom-K candidates under the same aggregation budget. All variants, including the PFT-lt baseline, are initialized from the same PFT-lt ×4 checkpoint and fine-tuned for 100K iterations under identical settings. As shown in Figure 5(a), full-window top-K improves PFT-lt by 0.08 dB on Set14 and 0.09 dB on Urban100, whereas random-K and bottom-K substantially degrade performance. Together, these results show that progressive inheritance can prematurely exclude candidates that become relevant at greater depth, and that restoring access to them improves reconstruction quality.

Table 3: Ablation study of EvRC-lt at $\times 4$ scale on Set14 and Urban100.

Variants	IR	DR	Set14		Urban100	
			PSNR	SSIM	PSNR	SSIM
Baseline			28.86	0.7872	26.91	0.8082
Immediate only	✓		28.95	0.7894	27.24	0.8172
Deferred only		✓	28.96	0.7897	27.19	0.8160
EvRC-lt (ours)	✓	✓	29.02	0.7906	27.31	0.8201

(a) Impact of component modules.

Retrieval Design	MACs	Set14		Urban100	
		PSNR	SSIM	PSNR	SSIM
Static Reader	72.5G	28.93	0.7888	27.15	0.8150
Query Conditioned	72.7G	28.96	0.7898	27.22	0.8164
EvRC-lt (ours)	72.8G	29.02	0.7906	27.31	0.8201
Cross Attention	**409.8G**	29.04	0.7909	27.40	0.8217

(b) Impact of Deferred Retrieval Designs.

Visualization of Response Recovery. Figure 5(b) visualizes the response contributions for target queries. PFT exposes only the sparse contributions available through the inherited support. Immediate Recovery reincorporates responses discarded at the current pruning stage, while Deferred Recovery further introduces responses carried forward from earlier stages. Together, they enrich the response with information from structurally related positions, without altering the pruning schedule.

Impact of Component Modules. Complementing the qualitative visualization above, Table 3a quantitatively examines the contribution of Immediate Recovery and Deferred Recovery. The baseline shares the same progressive backbone as EvRC-lt with both components removed. From the results we observe that adding Immediate Recovery alone yields consistent gains over the baseline (+0.09 dB on Set14, +0.33 dB on Urban100), confirming that fusing discarded response information back into the retained stream effectively compensates for the response information lost during pruning. Deferred Recovery independently provides notable improvements of +0.10 dB on Set14 and +0.28 dB on Urban100, respectively, demonstrating that compact traces of discarded responses remain useful to later layers and aggregating cross-layer evidence is effective. Combining both components yields the best results, with gains of +0.16 dB and +0.40 dB on Set14 and Urban100, respectively, indicating that the two components are complementary and mutually reinforcing.

Impact of Deferred Retrieval Designs. Table 3b studies how information is retrieved from the Trace Bank by comparing different retrieval designs. The static reader assigns a learned but input-independent weight to each response trace. A query-conditioned method improves over the static baseline by predicting retrieval weights from the current query (28.93→28.96 dB on Set14, 27.15→27.22 dB on Urban100), but relies solely on the query without accounting for the actual content of each trace. Our full design further incorporates the discarded attention mass and trace content, combining a mass prior with query–trace compatibility, which raises performance to 29.02 dB and 27.31 dB with only a marginal increase in MACs (72.7G→72.8G). This confirms that content-aware routing is more effective than query-only prediction. As a more expressive reference, we further allow each query to retrieve traces from all spatial positions through cross-attention. Relaxing spatial correspondence provides only marginal gains of $0.02/0.09$ dB on Set14/Urban100, while increasing computation from 72.8G to 409.8G MACs. This result suggests that spatially aligned retrieval captures most of the available benefit at substantially lower cost.

6 Conclusion

In this work, we present EvRC, a lightweight plug-in for progressive sparse attention that addresses the permanent loss of response information caused by pruning. The key insight is that pruning a candidate from the interaction support does not render its accumulated response useless — it simply removes the opportunity for future layers to access that information. EvRC recovers this opportunity through two complementary mechanisms: Immediate Recovery folds the discarded response into the current representation at pruning time, while Deferred Recovery deposits a compact, query-retrievable trace into a shared bank for use by subsequent layers.

Experiments on standard SR benchmarks demonstrate that EvRC consistently achieves state-of-the-art performance under both classical and lightweight settings. Ablation studies confirm that the two components address distinct aspects of information loss and are mutually reinforcing. We hope this work encourages a broader rethinking of progressive sparse architectures: rather than treating pruned tokens as waste, their residual information can be harvested at modest additional cost.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript writing, literature retrieval, and code development. Specifically, AI tools were used to improve the clarity, fluency, and academic tone of the text; to identify potentially related works; and primarily to implement and debug routine code components. All AI-assisted content was carefully reviewed by the authors: AI-polished text was revised to ensure technical accuracy and consistency with the original intent, and all code was thoroughly reviewed, tested, and validated for correctness. The authors take full responsibility for the final content of this work, including all text, claims, and methodology.

REPRODUCIBILITY STATEMENT

Section 4 describes EvRC and its integration into progressive sparse attention, while Appendix A provides the architecture details including configurations, pruning schedules, and placement of recovery modules. Section 5.1 specifies the datasets, degradation model, and evaluation protocol, and Appendix B details data preparation, training, and initialization. Appendix C describes the procedures used for the analysis experiments and ablation experiments. We will release the code and pretrained checkpoints upon acceptance.

REFERENCES

Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *European conference on computer vision*, pp. 256–272. Springer, 2018.

Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *British machine vision conference (BMVC)*, 2012.

Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12294–12305. IEEE, 2021.

Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22367–22377. IEEE, 2023a.

Zheng Chen, Yulun Zhang, Jinjin Gu, Linghe Kong, Xin Yuan, et al. Cross aggregation transformer for image restoration. *Advances in Neural Information Processing Systems*, 35:25478–25490, 2022.

Zheng Chen, Yulun Zhang, Jinjin Gu, Linghe Kong, Xiaokang Yang, and Fisher Yu. Dual aggregation transformer for image super-resolution. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12278–12287. IEEE, 2023b.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pp. 184–199. Springer, 2014.

Hang Guo, Yong Guo, Yaohua Zha, Yulun Zhang, Wenbo Li, Tao Dai, Shu-Tao Xia, and Yawei Li. Mambairv2: Attentive state space restoration. In *2025 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 28124–28133. IEEE, 2025.

Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5197–5206, 2015.

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1646–1654, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Dongheon Lee, Seokju Yun, and Youngmin Ro. Emulating self-attention with convolution for efficient image super-resolution. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24467–24477. IEEE, 2025.

Wenbo Li, Xin Lu, Shengju Qian, Jiangbo Lu, Xiangyu Zhang, and Jiaya Jia. On efficient transformer-based image pre-training for low-level vision. *arXiv preprint arXiv:2112.10175*, 2021.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *2021 IEEE/CVF international conference on computer vision workshops (ICCVW)*, pp. 1833–1844. IEEE, 2021.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pp. 1132–1140. IEEE, 2017.

Yifei Liu, Mathias Gehrig, Nico Messikommer, Marco Cannici, and Davide Scaramuzza. Revisiting token pruning for object detection and instance segmentation. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2646–2656. IEEE, 2024.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9992–10002. Ieee, 2021.

Wei Long, Xingyu Zhou, Leheng Zhang, and Shuhang Gu. Progressive focused transformer for single image super-resolution. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2279–2288. IEEE, 2025.

Xiaotong Luo, Yuan Xie, Yulun Zhang, Yanyun Qu, Cuihua Li, and Yun Fu. Latticenet: Towards lightweight image super-resolution with lattice block. In *European conference on computer vision*, pp. 272–289. Springer, 2020.

Xiaotong Luo, Zekun Ai, Qiuyuan Liang, Ding Liu, Yuan Xie, Yanyun Qu, and Yun Fu. Adaformer: Efficient transformer with adaptive token sparsification for image super-resolution. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 4009–4016, 2024.

David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, volume 2, pp. 416–423. Ieee, 2001.

Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia tools and applications*, 76(20):21811–21838, 2017.

Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image super-resolution with non-local sparse attention. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3516–3525. IEEE, 2021.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, Adam Lerer, et al. Automatic differentiation in pytorch. 2017.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

Jian-Nan Su, Min Gan, Guang-Yong Chen, Wenzhong Guo, and CL Philip Chen. High-similarity-pass attention for single image super-resolution. *IEEE Transactions on Image Processing*, 33: 610–624, 2024.

Zixun Sun, Yubo Dong, Hehe Fan, and Yi Yang. Hi-lo prune: Look at what you'll lose before pruning with hierarchical token selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31941–31951, 2026.

Yuchuan Tian, Hanting Chen, Chao Xu, and Yunhe Wang. Image processing gnn: Breaking rigidity in super-resolution. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24117. IEEE, 2024.

Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pp. 1110–1121. IEEE, 2017.

Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *2017 IEEE international conference on computer vision (ICCV)*, pp. 4809–4817. IEEE, 2017.

Dinh Phu Tran, Thao Do, Saad Wazir, Seongah Kim, Seon Kwon Kim, and Daeyoung Kim. Sat: Selective aggregation transformer for image super-resolution. *arXiv preprint arXiv:2604.07994*, 2026.

Hang Wang, Xuanhong Chen, Bingbing Ni, Yutian Liu, and Jinfan Liu. Omni aggregation networks for lightweight image super-resolution. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22378–22387. IEEE, 2023a.

Xiaohang Wang, Xuanhong Chen, Bingbing Ni, Hang Wang, Zhengyan Tong, and Yutian Liu. Deep arbitrary-scale image super-resolution via scale-equivariance pursuit. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1786–1795. IEEE, 2023b.

Yan Wang, Yi Liu, Shijie Zhao, Junlin Li, and Li Zhang. Camixersr: Only details need more "attention". In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25837–25846. IEEE, 2024.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10799–10808. IEEE, 2022.

Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pp. 711–730. Springer, 2010.

Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2808–2817. IEEE, 2017.

Leheng Zhang, Yawei Li, Xingyu Zhou, Xiaorui Zhao, and Shuhang Gu. Transcending the limit of local window: Advanced super-resolution transformer with adaptive token dictionary. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2856–2865. IEEE, 2024.

Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. Efficient long-range attention network for image super-resolution. In *European conference on computer vision*, pp. 649–667. Springer, 2022.

Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *European conference on computer vision*, pp. 294–310. Springer, 2018a.

Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 2472–2481. IEEE, 2018b.

YiGang Zhao, Chaowei Zheng, JianNan Su, and GuangYong Chen. Efficient learnable collaborative attention for single-image super-resolution. *Journal of Electronic Imaging*, 33(6):063005–063005, 2024.

A Architecture Details

Backbone configuration. Both EvRC and EvRC-lt operate on the low-resolution feature grid until the reconstruction head, without spatial resampling in the Transformer body. A 3×3 convolution first extracts shallow features, which are processed by a sequence of residual Transformer groups. Following SwinIR (Liang et al., 2021), each group alternates regular and shifted-window attention blocks and ends with a 3×3 convolution and a group-level residual connection. Each block contains a self-attention sublayer and a convolutional feed-forward sublayer. Attention is initially dense and becomes sparse after support contraction. At each pruning stage, the current-layer attention weights are used directly for top-K selection, without an auxiliary selector or PFT's cross-layer Focus modulation. For a 32×32 window, shifted-window layers use a cyclic shift of 16 pixels.

Table 4 summarizes the two model configurations. The convolutional feed-forward network and local positional encoding both use a kernel size of 7. The support schedule gives the candidate-set size within each of the regular and shifted-window lineages. Candidate indices are propagated independently along the regular and shifted lineages. At each pruning event, top-k selection partitions the incoming candidates into retained and discarded bands.

Table 4: Architecture configurations of the classical and lightweight models.

Configuration	EvRC	EvRC-lt
Residual Transformer groups	6	5
Group depths	$(4, 4, 4, 6, 6, 6)$	$(2, 4, 6, 6, 6)$
Total attention layers	30	24
Feature dimension C	240	52
Attention heads H	6	4
Head dimension d	40	13
FFN expansion ratio	2	1
Window / shift size	$32/16$	$32/16$
Support schedule	$(1024, 256, 128, 64, 32, 16)$	$(1024, 256, 128, 64, 32)$
Trace dimension d_m	8	8
Traces per lineage	5	4

Placement of recovery modules. Immediate Recovery is applied only at pruning layers. For Deferred Recovery, Table 5 reports the exact module placement using global, zero-based attention-layer indices. Every write layer is a pruning layer and also applies Immediate Recovery. Starting from the second pruning event of each lineage, the layer reads all traces produced by its preceding events before writing its own trace. After the final pruning event, one additional reader is placed at the first subsequent layer.

Reconstruction heads. After the Transformer body, a final normalization and a 3×3 convolution are followed by a long residual connection to the shallow features. The classical model first projects the backbone features to 64 channels, performs upsampling in the feature space using standard PixelShuffle blocks, and finally maps the upsampled features to RGB. The lightweight model uses a single 3×3 convolution to predict $3s^2$ sub-pixel channels, followed by PixelShuffle at scale s. During inference, inputs are reflection-padded to a multiple of the window size and the reconstructed output is cropped to the requested spatial dimensions.

More details on trace reading. The routing function in Eq. 9 is implemented as follows, with stage and head indices omitted:

$$w = G(\rho, s) = \lambda\, \sigma(\mathrm{logit}(\bar{\rho}) + b + \eta \tanh s)\,, \tag{13}$$

where $\bar{\rho}$ is the discarded-mass fraction clipped to $(0, 1)$, s is the query–trace compatibility, and b, η, and λ are learned scalars. The logit places the mass on a scale where the bias and similarity term can

Table 5: Trace-writing and trace-reading layers. R and S denote the regular and shifted-window lineages, respectively.

Model	Lineage	Write layers	Read layers
EvRC	R	$\{4, 8, 12, 18, 24\}$	$\{8, 12, 18, 24, 26\}$
	S	$\{5, 9, 13, 19, 25\}$	$\{9, 13, 19, 25, 27\}$
EvRC-lt	R	$\{2, 6, 12, 18\}$	$\{6, 12, 18, 20\}$
	S	$\{3, 7, 13, 19\}$	$\{7, 13, 19, 21\}$

adjust it additively. The sigmoid maps the adjusted score back to a gate in $(0, 1)$: when $b = s = 0$, this gate equals $\bar{\rho}$. Thus, discarded mass provides a reference weight that query–trace compatibility can increase or decrease, while the unconstrained λ allows the final contribution w to be signed.

Trace Bank. The Trace Bank is reset at the start of each forward pass, with separate states for regular and shifted-window lineages. Each entry stores a mass scalar and a d_m-dimensional trace per query and source head. Traces are kept in canonical image coordinates to preserve query alignment across window shifts. Bank states remain differentiable during training.

B Training Details

Data preparation. Following previous works (Chen et al., 2023a; Liang et al., 2021), our classical models are trained on DF2K, and lightweight models are trained on DIV2K. For an upscaling factor s, we randomly crop a 64×64 patch from the LR image and the corresponding $(64s) \times (64s)$ patch from its HR counterpart. The paired patches are augmented using random horizontal and vertical flips and rotations by multiples of 90°. The models are trained on two NVIDIA H20 GPUs.

Optimization. Training minimizes the pixel-wise ℓ_1 distance between the reconstructed and ground-truth RGB images. We use Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and an initial learning rate of 2×10^{-4}. Each model is optimized for 1000K iterations, and the learning rate is reduced by a factor of 0.5 every 200K iterations. The global mini-batch contains 32 LR patches.

Parameter initialization. For Immediate Recovery, the learnable coefficient α is initialized to one, so the initial response includes the complete incoming-support response. For Deferred Recovery, the output gain β is initialized to zero, so Deferred Recovery contributes no correction initially and is activated only if beneficial. Since deeper readers aggregate an increasing number of traces, the coefficients λ are initialized to $1/\sqrt{|\mathcal{R}|}$ to keep the variance of the aggregated response approximately independent of the number of available traces.

Matched retraining of PFT-lt. To separate architectural differences from training-recipe differences, we retrain PFT-lt at $\times 4$ using the same protocol as EvRC-lt, including the training data, optimizer, learning-rate schedule and number of iterations. Table 6 reports the resulting comparison. EvRC-lt consistently outperforms the retrained PFT-lt across all five benchmarks.

Table 6: Controlled comparison at $\times 4$ under the same training and evaluation protocol.

Model	Set5		Set14		B100		Urban100		Manga109	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
PFT-lt (retrained)	32.61	0.9006	28.94	0.7895	27.79	0.7443	27.21	0.8180	31.45	0.9196
EvRC-lt	**32.63**	**0.9007**	**29.02**	**0.7906**	**27.83**	**0.7455**	**27.31**	**0.8201**	**31.70**	**0.9222**

C Details of Analysis Experiments

Response visualization in Figure 1 and Figure 5b. We fix the image, 32×32 window, target query, attention head, and target layer across all response maps. For full attention and progressive sparse attention, each location j is colored by the ℓ_2 norm of its direct response contribution,

$\|a_j v_j\|_2$, with contributions outside the active support set to zero. The Immediate Recovery map additionally includes $\alpha a_j v_j$ for candidates discarded at the current pruning stage.

Deferred Recovery does not recompute attention over earlier key positions. For visualization, we decompose each written trace into its source-position components, $z^r = \sum_j z_j^r$, and propagate each component through the actual reader while keeping the learned routing weights fixed:

$$\delta y_j^t = \sum_{r \in \mathcal{R}^t} \mathcal{G}_t^r(z_j^r), \qquad \sum_j \delta y_j^t = \Delta y^t, \tag{14}$$

where $\mathcal{G}_t^r$ denotes the corresponding gated decoding operation. The Deferred Recovery map visualizes the resulting contribution at each source position after it is added to the current response. It therefore represents historical response provenance rather than newly computed attention to those positions.

Key-selection intervention in Figure 5a. We retain the PFT-lt $\times 4$ architecture and its per-layer aggregation budget, but vary the candidate pool and selection rule. The progressive baseline inherits the retained support, whereas the three full-window variants recompute attention over all valid keys at every layer and select the highest-scoring, lowest-scoring, or uniformly sampled K keys, respectively. PFT's cross-layer Focus modulation is retained. For the full-window variants, we propagate the complete attention map before selection, separately for regular and shifted-window layers. Each layer multiplies its current attention weights by the preceding map from the same lineage and normalizes the result before selection. Keeping these maps indexed by window positions preserves correspondence between historical weights and keys even when the selected set changes. All full-attention variants aggregate the selected values using their corresponding Focus-modulated weights. All four variants, including the progressive baseline, are initialized from the same PFT-lt checkpoint and fine-tuned for 100K iterations for a fair comparison. These full-window variants are used only for analysis, and their additional QK computation is not part of EvRC's inference cost.

Retrieval design comparison in Table 3b. For the static and query-conditioned controls, a linear writer compresses the discarded response directly to d_m dimensions, without separately storing its mass and conditional-response residual. The static reader uses one learned scalar for each available stage, source head, and target head; it has no query-processing layer. The query-conditioned method instead applies LayerNorm and a single linear layer $\mathbb{R}^d \to \mathbb{R}^{|\mathcal{R}|H}$ to the current query of each target head, producing one routing logit per available stage and source head. A sigmoid converts these logits to gates. Neither control computes query–trace compatibility or uses a separate mass prior. Both aggregate the weighted traces by source head and use the same linear output decoder as EvRC.

We additionally evaluate a cross-position retrieval variant to examine whether relaxing the spatial correspondence of trace retrieval provides further benefit. While the preceding readers access only historical traces associated with the same spatial query, this variant allows each query to retrieve traces from all spatial positions within the window. It projects the target-head query and the stored traces into the d_m-dimensional space, treats traces from all available stages, source heads, and window positions as the memory, and aggregates them using scaled dot-product attention. The resulting source-head features are concatenated and decoded through the same $\mathbb{R}^{Hd_m} \to \mathbb{R}^d$ output projection. This cross-position variant is included as a more expressive retrieval reference rather than a matched comparison of other designs. Its additional MACs arise from both dot-product attention and the enlarged spatial retrieval scope. It therefore measures whether the modest benefit of cross-position retrieval justifies its substantially higher computational cost.

D Efficiency Analysis

Computational complexity. We take PFT (Long et al., 2025) as a comparison. Consider a window with N queries and feature dimension C. At pruning stage r, PFT scores k_{in} candidate keys per query and aggregates k_{out} retained values, giving an attention cost of $\mathcal{O}(NC(k_{\text{in}} + k_{\text{out}})) \approx \mathcal{O}(NCk_{\text{in}})$ when $k_{\text{in}} \gg k_{\text{out}}$. EvRC leaves the candidate set and query–key scoring unchanged. It additionally aggregates values over the incoming support at a cost of NCk_{in} MACs. Immediate Recovery and trace writing share this response. With H heads, trace dimension d_m, and R available traces, writing and reading add $\mathcal{O}(NCd_m)$ and $\mathcal{O}(NHCd_m + NRH^2d_m)$, respectively. The Trace

Table 7: Inference efficiency at $\times 4$ scale. The results are calculated on 1280×720 HR outputs.

Setting	Model	Params	MACs	Peak memory	Latency
Classical	PFT	19.8M	1.41T	7.3GB	834ms
	EvRC	19.9M	1.46T	8.8GB	870ms
Lightweight	PFT-lt	792K	63.7G	4.2GB	392ms
	EvRC-lt	813K	72.8G	5.5GB	397ms

Bank stores $\mathcal{O}(NRH(d_m + 1))$ values. Since H, d_m, and R are fixed and small in our models, a sparse stage thus retains $\mathcal{O}(NCk_{\text{in}})$ scaling, with additional constant-factor and linear-in-N costs. The initial dense stage remains $\mathcal{O}(N^2C)$ for both methods.

Measured inference cost. Table 7 compares EvRC model with PFT model under both classical and lightweight settings. We measure full-image, batch-one FP32 inference for a 1280×720 HR output on an H20 GPU, excluding image loading and preprocessing. Note that the MACs of PFT (Long et al., 2025) and ATD (Zhang et al., 2024) reported in Table 1 and Table 2 are recomputed at the corresponding resolution to ensure a fair comparison with other methods. Latency is averaged over 10 timed runs after 10 warm-up runs; peak memory is measured using `torch.cuda.max_memory_allocated`.

E Limitations

EvRC preserves information from pruned interactions as compact response traces, but does not restore the individual interactions for exact re-evaluation at deeper layers. Therefore, this is an approximate compensation for evicted evidence, rather than an exact restoration. Whether higher-fidelity recovery mechanisms could yield further gains remains an open question. Additionally, EvRC is designed and evaluated specifically for image super-resolution; its generalization to other tasks built on progressive sparse attention remains to be explored.

F More Visual Results

Figures 6 and 7 provide additional visual comparisons on Urban100 and Manga109. All methods are shown using the same input and crop for each example. These images contain repeated lines, dense geometric patterns, and fine illustration contours, for which small reconstruction errors are readily visible. Compared with the competing methods, EvRC better preserves the continuity and alignment of these structures, demonstrating the strengths of our approach.

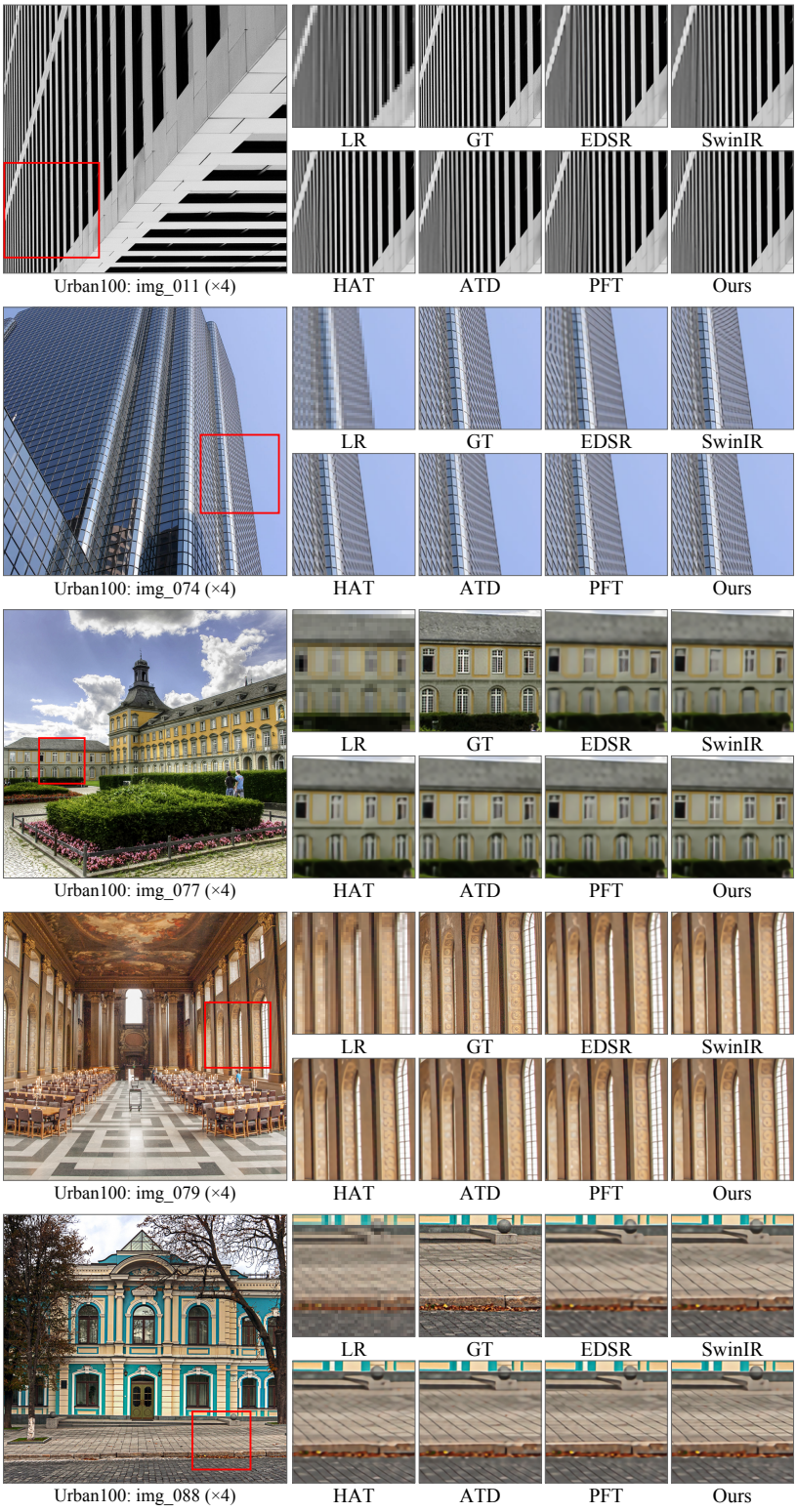


Figure 6: Additional visual comparisons on Urban100.



Figure 7: Additional visual comparisons on Manga109.