Linear Recurrent Unit with Semantic Modulation for Image Super-Resolution

Supplementary Material

A. Training Settings

Classic SR Following previous works [5, 6], we use DIV2K [9] and Flickr2K [6] as the training datasets. We train with batch size 32. Patches are augmented by random flips and 90°, 180°, 270° rotations. Training proceeds in two steps. In the first step, inputs are cropped to 64×64, and we minimize the ℓ_1 pixel loss using AdamW [7] with $\beta_1 = 0.9$, $\beta_2 = 0.9$. For $\times 2$ upscaling, training runs for 300k iterations with initial learning rate 2×10^{-4}, halved at the 250k milestone. In the subsequent fine-tuning step, following previous work [11], we use larger patches (96×96 for LSM-S, 92×92 for LSM) chosen for NVIDIA RTX 3090 GPU capacity to better exploit the semantic modulating unit (SMU) and memory efficiency. The training runs for 200k iterations and the same initial learning rate is used with halving at milestones. Total training is 500k iterations. For $\times 3$ and $\times 4$, we skip first step for efficiency, initialize from $\times 2$ weights, and apply only fine-tuning step for 250k iterations. A 10k warm-up at each step increases the learning rate linearly from 0 to the initial value.

Lightweight SR In the LSM-light model, only the DIV2K [9] dataset is used for training unlike the classic SR. To match the batch size with previous works [3, 10, 12], we doubled it compared to the classic SR setting, while keeping all other training strategies identical to those of LSM-S.

B. Additional Quantitative Comparison

Our objective is to propose an efficient SR backbone based on LRU, a lightweight SSM variant. To this end, all models were trained under practical compute constraints, using 24GB of GPU memory across 8 GPUs. Accordingly, the main paper primarily compares our model with existing small size baselines that adopt Transformer and Mamba backbones. We further demonstrate the potential of our model as a new SR backbone by evaluating it on larger models and higher-resolution datasets in terms of performance and efficiency.

Comparison with Large Models We compare our model with two recent larger models on $\times 4$ SR: Transformer-based HAT [1] and Mamba-based MambaIRv2-B [3]. As shown in Tab. B.1, our model achieves competitive performance despite reducing the number of parameters and FLOPs significantly by 38% and 36% compared to HAT, and by 44% and 42% compared to MambaIRv2-B, respectively.

Comparison with Dictionary-based Model ATD [11], which inspired our approach, employs category-based attention with a parallel architecture. While minimizing parameter overhead, it achieves strong performance gains. The dictionary operation used in ATD plays a key role in overcoming the spatial limitations of local attention. Similarly, we reinterpret the mechanism by mitigating the single-scan limitation of LRU in our model. Futhermore we assign it the role of computationally efficient modulation. We compare the ATD-light and LSM-light models at the $\times 2$ scale in Tab. B.2. Despite a similar number of parameters due to the parallel structure of ATD, our model reduces FLOPs by 35% and achieves 1.5× lower latency, while maintaining comparable performance. This result supports the validity of our approach, where the integration of dynamic modulation enhances suitability for SR tasks by extending long-range modeling capacity of LRU.

Table B.1. Quantitative comparison with large size models

Method	# Params	FLOPs	Set5	Set14	B100	Ub.100	Mg.109
HAT	20.8M	412G	33.04	29.23	**28.00**	**27.97**	32.48
MambaIRv2-B	23.1M	455G	**33.14**	29.23	**28.00**	27.89	**32.57**
LSM	**12.9M**	**265G**	32.96	**29.24**	**28.00**	27.94	32.42

Table B.2. Quantitative comparison with dictionary-based model

Method	Latency	# Params	FLOPs	Metric	Set5	Set14	B100	Ub.100	Mg.109
ATD-light	914ms	**753K**	380G	PSNR	**38.29**	34.10	**32.39**	**33.27**	**39.52**
				SSIM	**0.9616**	0.9217	**0.9023**	0.9375	**0.9789**
LSM-light	**611ms**	763K	**282G**	PSNR	38.27	**34.14**	**32.39**	33.24	39.35
				SSIM	0.9615	**0.9219**	**0.9023**	**0.9379**	0.9784

Table B.3. Quantitative comparison on high-resolution datasets

Method	# Params	FLOPs	Test2k		Test4k		Test8k	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SwinIR	11.9M	215.3G	27.99	0.7898	29.48	0.8349	35.57	0.9034
MambaIRv2-S	**9.8M**	**202.9G**	28.07	0.7909	29.56	0.8359	**35.74**	0.9047
LSM-S	**9.9M**	**203.5G**	**28.11**	**0.7924**	**29.60**	**0.8371**	35.73	**0.9050**

Comparison on high-resolution datasets Our work emphasizes that the carefully designed initialization of the LRU provides a foundational basis for its recurrence behavior, which is essential for effective long-range modeling. To further validate this claim, we conduct additional experiments on high-resolution datasets [4]. As shown in Tab. B.3, our LSM-S consistently outperforms SwinIR [5] and MambaIRv2-S [3] across most datasets despite its efficient computational complexity.

C. Preliminaries

LRUs [8] serve as the core backbone of this study. Unlike conventional RNNs that struggle with long-sequence learning due to vanishing and exploding gradient problems and the inefficiency of sequential computation, LRUs demonstrate strong performance in long-range dependency modeling and high computational efficiency through a series of structural changes and initialization strategies. We provide a full derivation of the LRU formulation presented in the methodology section of the main paper.

Vanilla RNN A standard RNN layer [2] consumes an H_{in}-dimensional input, produces an N-dimensional hidden state and an H_{out}-dimensional output, and typically includes a non-linear activation function σ:

$$\begin{aligned} h_k &= \sigma(Ah_{k-1} + Bu_k), \\ y_k &= Ch_k + Du_k, \end{aligned} \tag{C.1}$$

where $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times H_{\text{in}}}$, $C \in \mathbb{R}^{H_{\text{out}}\times N}$, and $D \in \mathbb{R}^{H_{\text{out}}\times H_{\text{in}}}$ are trainable, and $h_0 = 0$.

Linearizing Recurrences The first key modification in LRU is the removal of the non-linearity σ from the hidden state update, opting for a linear recurrence. This enhances learning stability and enables parallelization without sacrificing model expressivity. The overall non-linearity is instead provided by Multi-Layer Perceptron (MLP) or Gated Linear Unit (GLU) blocks placed between each LRU block:

$$\begin{aligned} h_k &= Ah_{k-1} + Bu_k, \\ y_k &= Ch_k + Du_k, \end{aligned} \tag{C.2}$$

which unrolls as $h_k = A^k h_0 + \sum_{j=0}^{k-1} A^j B\, u_{k-j}$. In long sequences, the hidden state can explode or vanish depending on the magnitude of the eigenvalues of matrix A.

Complex Diagonal Recurrences To maximize the computational efficiency of the linear recurrence, matrix A is reparameterized as a complex-valued diagonal matrix Λ. This leverages the eigendecomposition of A, expressed as $A = P\Lambda P^{-1}$. In the eigen-basis $\bar{h}_k = P^{-1}h_k$, the hidden state can be linearly expressed as:

$$\begin{aligned} \bar{h}_k &= \Lambda\bar{h}_{k-1} + \bar{B}\, u_k, \\ \bar{y}_k &= \bar{C}\, \bar{h}_k + Du_k, \end{aligned} \tag{C.3}$$

where $\bar{B} = P^{-1}B$ and $\bar{C} = CP$. Then $\bar{h}_k = \Lambda^k \bar{h}_0 + \sum_{m=0}^{k-1} \Lambda^m \bar{B}\, u_{k-m}$ with elementwise powers on the diagonal of Λ, which is parallel-scan friendly.

Stable Exponential Parameterization LRU enhances learning stability and strengthens long-range dependency modeling by controlling the eigenvalue λ distribution of the recurrent matrix, rather than relying on a specific deterministic initialization. Exponential parameterization is used to control the magnitude and phase of eigenvalues, which effectively separates them to improve the performance of optimizers:

$$\begin{aligned} \Lambda &= \operatorname{diag}(\lambda), \\ \lambda_j &= \exp\big(-\exp(\nu_j^{\log})\big)\ \exp\big(i\,\exp(\theta_j^{\log})\big), \end{aligned} \tag{C.4}$$

where j refers to the index of each individual eigenvalue λ_j, which comes with trainable $\nu_j^{\log}$, $\theta_j^{\log} \in \mathbb{R}$. For initialization, λ_j are sampled to be uniformly distributed on an annulus in the complex plane, defined by inner radius $r_{\min}$ and outer radius $r_{\max}$. The phase of λ_j is uniformly sampled within a specified range, typically $[0, 2\pi]$ or a smaller slice for tasks requiring very long-range reasoning. Specifically, the trainable parameters $\nu_j^{\log}$ and $\theta_j^{\log}$ are initialized using independent uniform random variables u_1, $u_2 \in [0, 1]$ as follows:

$$\begin{aligned} \nu_j^{\log} &= \log\left(-\frac{1}{2}\log\left(u_1(r_{\max}^2 - r_{\min}^2) + r_{\min}^2\right)\right), \\ \theta_j^{\log} &= \log(\theta_{max} u_2), \end{aligned} \tag{C.5}$$

where θ_{max} defines the upper limit of the phase sampling range. This initialization strategy sets an effective dependency range for each λ_j and the results are further analyzed in the ablation section of the main paper.

Normalization To prevent hidden activation blow-up when $|\lambda_j|$ is close to one, LRU introduces a forward normalization factor $\gamma_j = \sqrt{1 - |\lambda_j|^2}$ applied channelwise. The modified hidden state update and output equations are as follows:

$$\begin{aligned} \bar{h}_k &= \operatorname{diag}(\lambda) \odot \bar{h}_{k-1} + \gamma \odot (\bar{B}\, u_k), \\ \bar{y}_k &= \bar{C}\, \bar{h}_k + Du_k, \end{aligned} \tag{C.6}$$

where $\gamma = \operatorname{diag}(\gamma_j)$ broadcasts across channels and $\odot$ denotes elementwise multiplication.

D. Additional Visual Results

To further support the findings presented in the main paper, we provide additional qualitative visualizations.

Visualization of hidden states We further visualize the modulation effects on hidden states in Fig. D.1 to demonstrate the consistency of our findings. Following the same strategy, we sort channels across different models based on cosine similarity to highlight similarly activated responses. The results show that the vanilla LRU struggles to capture key textures such as the bird's beak and window patterns. With semantic categorization, hidden states exhibit more coherent activation across spatially distant pixels with similar meanings. Finally, applying the modulated LRU to categorized pixels allows the model to balance long-range semantic consistency and local texture, yielding the most faithful representations among variants.

Visualization of categorization We visualize the categorization results before feeding into the LRU in Fig. D.2.

Qualitative comparison Our LSM consistently reconstructs both semantic structures and fine-grained textures across a wide range of images. As shown in Fig. D.3, our method recovers structured patterns such as straight lines and architectural details with higher fidelity than competing models on the Urban100 dataset. In addition, in Fig. D.4, our approach effectively reconstructs irregular curved textures across datasets while minimizing artifacts that deviate from the ground-truth structure.

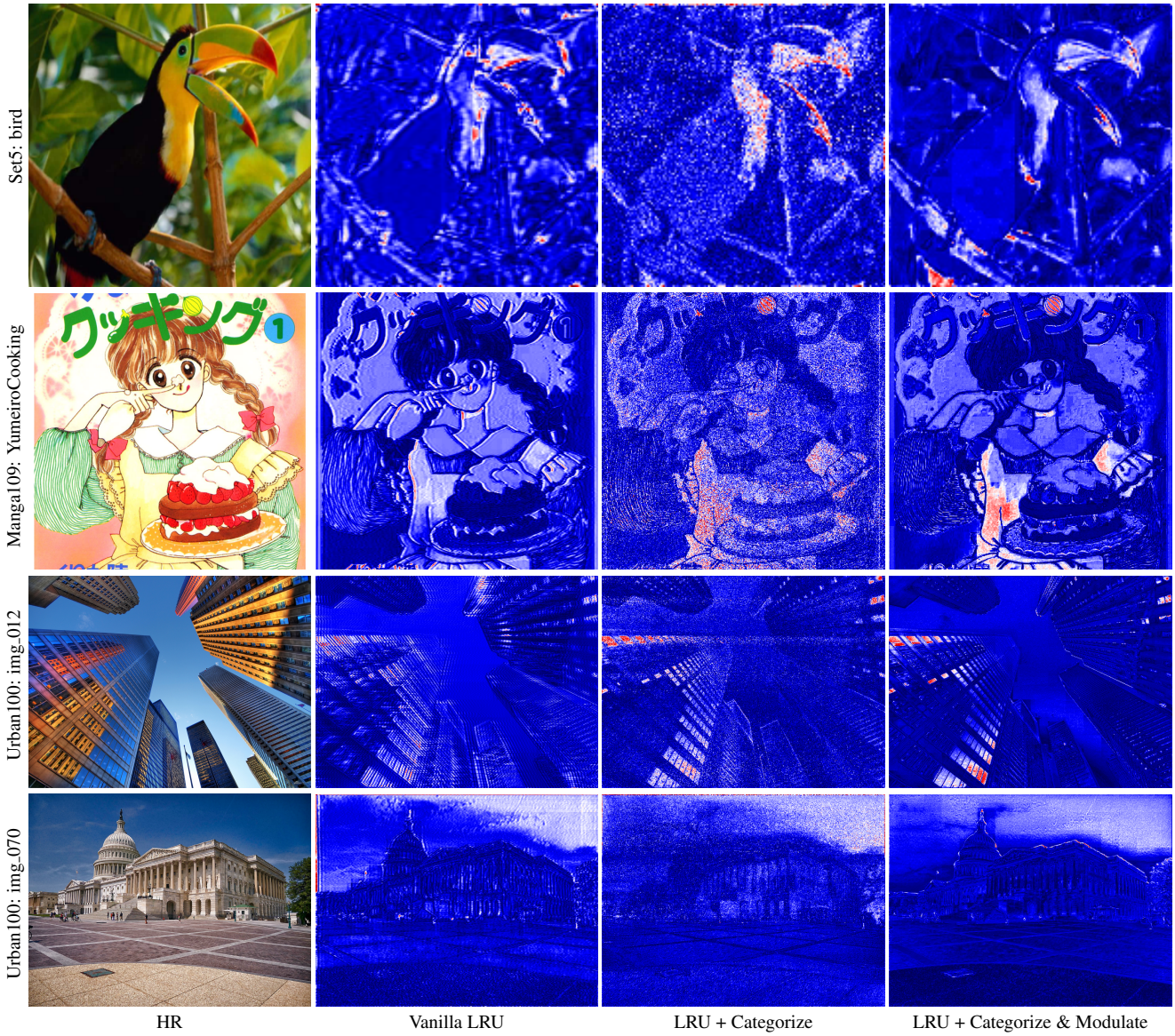

Figure D.1. Visualization of hidden states.

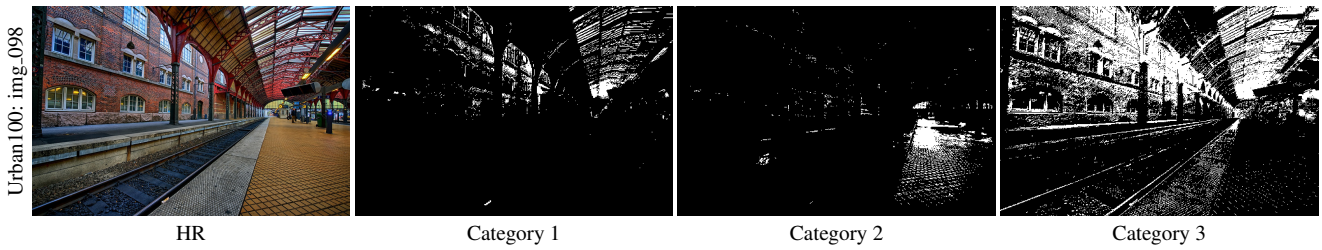

Figure D.2. Visualization of categorization results.

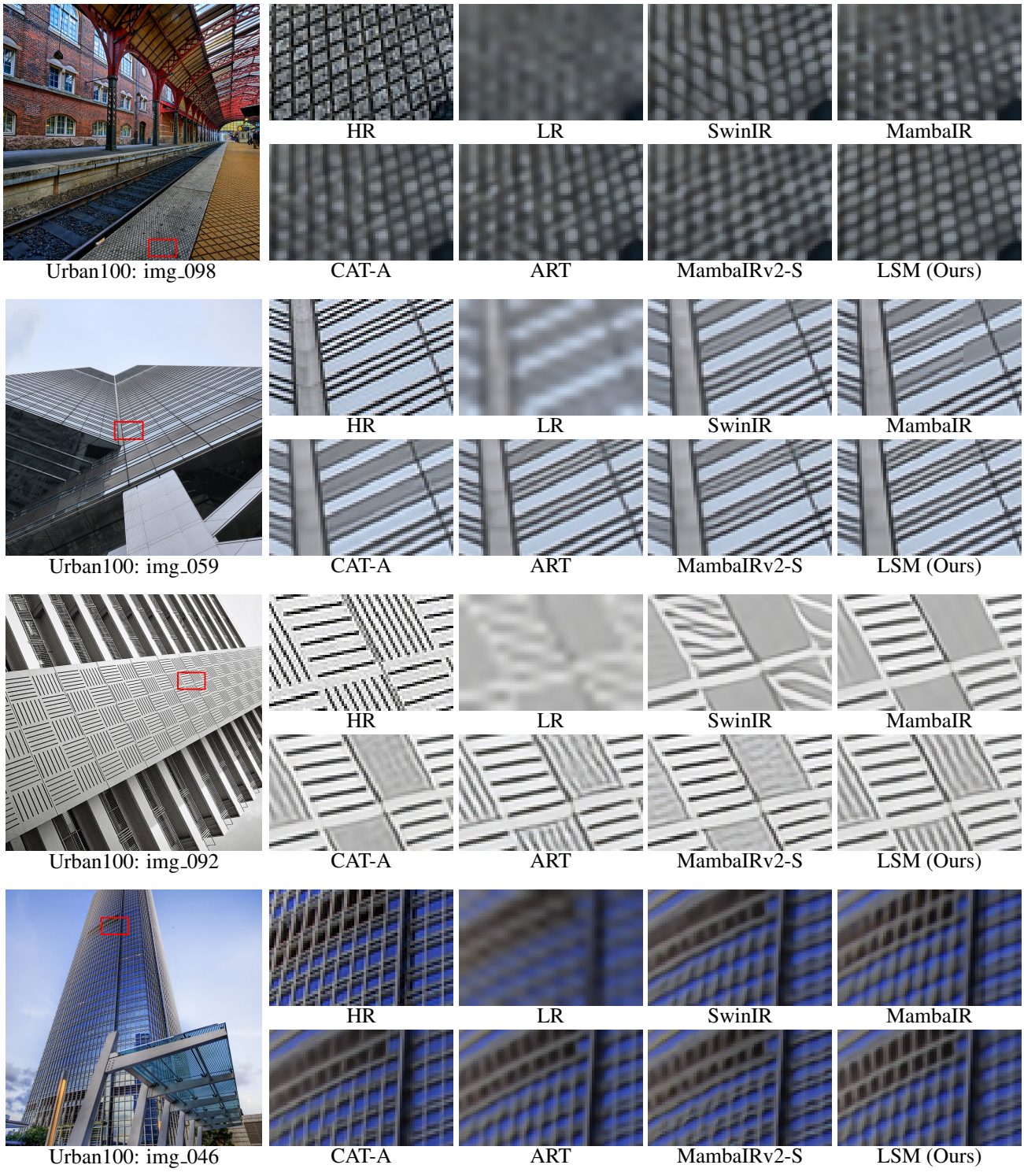

Figure D.3. Qualitative comparisons with competitive methods on $\times 4$ classic SR focusing on straight patterns.

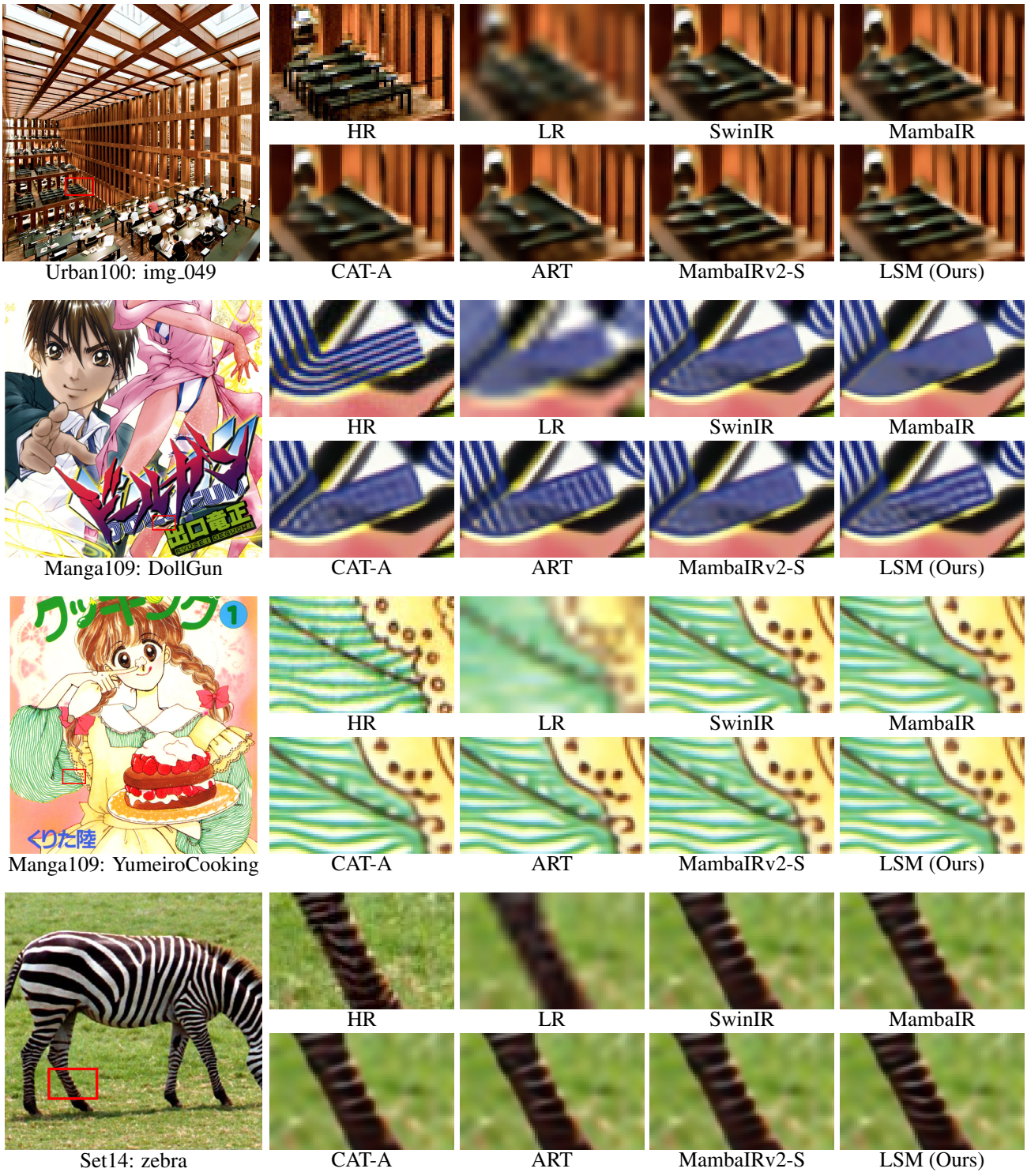

Figure D.4. Qualitative comparisons with competitive methods on $\times 4$ classic SR focusing on irregular and curved textures.

References

[1] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22367–22377, 2023. 1

[2] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 2

[3] Hang Guo, Yong Guo, Yaohua Zha, Yulun Zhang, Wenbo Li, Tao Dai, Shu-Tao Xia, and Yawei Li. Mambairv2: Attentive state space restoration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28124–28133, 2025. 1

[4] Xiangtao Kong, Hengyuan Zhao, Yu Qiao, and Chao Dong. Classsr: A general framework to accelerate super-resolution networks by data characteristic. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12016–12025, 2021. 1

[5] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 1

[6] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 1

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[8] Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pages 26670–26698. PMLR, 2023. 1

[9] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017. 1

[10] Hang Wang, Xuanhong Chen, Bingbing Ni, Yutian Liu, and Jinfan Liu. Omni aggregation networks for lightweight image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22378–22387, 2023. 1

[11] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2856–2865, 2024. 1

[12] Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. Efficient long-range attention network for image super-resolution. In *European conference on computer vision*, pages 649–667. Springer, 2022. 1