DIRECTSR: RETHINKING ZERO-ORIGIN RESIDUAL FLOW VIA OBSERVATION-GUIDED ENDPOINT PREDICTION FOR REMOTE SENSING SUPER-RESOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Remote sensing super-resolution requires both fine-detail recovery and strict spatial consistency with the observed scene. We revisit a zero-origin deterministic residual flow with target $R = x_{\mathrm{HR}} - B$ and linear path $x_t = tR$, for which the analytical velocity target remains constant and equal to R along the trajectory. This motivates direct endpoint recovery without explicit time-dependent residual transport. We propose **DirectSR**, which predicts the target residual through a single endpoint-predictor evaluation, conditioned on frozen multi-scale DINOv2 features, pixel-aligned differential responses, and an LR-consistency correction. We further study an optional candidate-aware refinement stage, **DirectSR+**. On Potsdam and Toronto $2\times$, DirectSR improves over a matched four-step residual flow by 2.4442 and 3.2632 dB, respectively. Under our unified evaluation protocol, DirectSR achieves the best PSNR, SSIM, and structural fidelity among the methods in the main comparison on Toronto $4\times$, while DirectSR+ achieves the best PSNR and SSIM on Potsdam and Toronto $2\times$. Final-candidate experiments further show that Stage-2 refinement is dataset dependent, supporting DirectSR Stage-1 as the default model. These results support direct endpoint recovery as a simpler alternative to iterative integration in the deterministic paired residual formulation studied here, without extending this conclusion to stochastic flow, diffusion, or multimodal generative models.

1 INTRODUCTION

Single-image super-resolution (SR) reconstructs a high-resolution (HR) image from a low-resolution (LR) observation. For remote sensing imagery, the challenge extends beyond generating sharp textures: building contours, road boundaries, and thin structures should remain spatially aligned with the underlying scene. This becomes particularly difficult at $4\times$ magnification, where substantial boundary and high-frequency information is absent from the LR input. A visually sharper reconstruction is therefore not necessarily a more accurate one, motivating joint consideration of reconstruction fidelity, structural alignment, and perceptual quality (Blau & Michaeli, 2018).

Learning-based SR has progressed along deterministic and generative directions. Deterministic methods such as EDSR, RCAN, and SwinIR directly learn LR-to-HR mappings and provide strong distortion-oriented reconstruction (Lim et al., 2017; Zhang et al., 2018; Liang et al., 2021). Diffusion- and flow-based approaches instead introduce generative processes for recovering missing high-frequency content, with recent methods exploring efficient diffusion, residual generation, one-step restoration, and semantic-guided Flow Matching (Meng et al., 2024; Yue et al., 2023; Wang et al., 2024; Gong et al., 2025). While these studies demonstrate the value of generative priors, the ill-posed nature of SR does not by itself imply that an iterative trajectory is always necessary. For paired remote sensing SR, where observation-aligned fidelity is a primary objective, it is therefore useful to ask whether a deterministic endpoint can be recovered more directly.

We revisit this question through a zero-origin deterministic residual formulation. Given the bicubic base $B = U_s(y)$, the missing HR information is represented as $R = x_{\mathrm{HR}} - B$. For the linear path $x_t = tR$, the analytical velocity target is $v^* = R$ throughout the path. This does not imply that a learned vector field is independent of time or state, but it shows that the ideal supervision al-

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.

ways points toward the same endpoint residual. This observation motivates an empirical alternative: directly predict the endpoint instead of recovering it through iterative residual transport.

These two perspectives jointly lead to **DirectSR**. The model uses the bicubic image as an observation-aligned reconstruction base and directly recovers the HR residual through a single pixel-space prediction. Direct prediction, however, does not imply unconstrained pixel regression: frozen multi-scale DINOv2 features provide semantic context for *what* structures are likely to be present, while pixel-aligned gradient, Laplacian, and directional high-pass responses provide local structural anchors for *where* observable transitions occur. An LR-consistency correction further preserves information already supported by the LR observation. Residual prediction and single-pass inference are not themselves new. Our contribution is to formulate endpoint residual prediction as a matched alternative to the studied zero-origin residual Flow and to combine it with observation-aligned semantic context, fixed differential cues, and LR consistency. The recovery formulation determines how the missing information is predicted, while the conditioning constrains what details are introduced and where they are placed.

We evaluate the proposed framework on Potsdam, Toronto, and AID under both $2\times$ and $4\times$ SR. In matched $2\times$ controls, DirectSR improves over the four-step residual Flow by 2.4442 dB on Potsdam and 3.2632 dB on Toronto. Under the primary $4\times$ setting, the default DirectSR model achieves the best PSNR and SSIM among the methods in the main Toronto comparison and ranks first on all structural metrics reported in the corresponding main comparison while remaining close to the best reconstruction result on Potsdam. Multi-seed experiments show a consistent positive contribution from Stage-1 semantic conditioning in the tested settings, whereas the benefit of Stage 2 refinement is dataset dependent. We therefore retain DirectSR Stage 1 as the default reconstruction model.

Our main contributions are:

1. We revisit a zero-origin deterministic residual formulation and show that direct endpoint recovery outperforms its matched trajectory-based counterpart in the evaluated paired SR settings.
2. We propose **DirectSR**, combining single-pass residual prediction with frozen DINOv2 semantics, pixel-aligned differential cues, and LR-consistent reconstruction for observation-aligned remote sensing SR.
3. We systematically validate DirectSR through matched Flow controls and multi-seed ablations, while showing that optional Stage 2 refinement is dataset dependent.

2 Related Work

Direct reconstruction and prediction targets. Direct LR-to-HR reconstruction has been extensively studied, from SRCNN and residual networks such as VDSR and EDSR to attention- and Transformer-based models including RCAN, SwinIR, and HAT (Dong et al., 2014; Kim et al., 2016; Lim et al., 2017; Zhang et al., 2018; Liang et al., 2021; Chen et al., 2023). Accordingly, DirectSR does not claim residual prediction, pixel-space restoration, or single-pass inference as novel. More relevant to our formulation is *what should be predicted directly*. *Back to Basics: Let Denoising Generative Models Denoise* revisits diffusion prediction targets and highlights the distinction between clean-data and noised-quantity prediction (Li & He, 2026). We adopt this perspective only at the formulation level: before introducing an elaborate transport process, it is worth asking whether the target admits a simpler parameterization. Unlike predicting clean images from stochastic noise, we study whether a uniquely paired HR residual in SR requires a time-dependent trajectory.

Diffusion, flow, and generative super-resolution. Diffusion models, Flow Matching, and Rectified Flow model restoration through stochastic or continuous transport processes (Lipman et al., 2023; Liu et al., 2023). Recent SR methods increasingly redesign these processes for efficient restoration: ResShift introduces residual-shifting diffusion, while SinSR distills restoration into a single-step model (Yue et al., 2023; Wang et al., 2024). For remote sensing SR, EDiffSR and FastDiffSR study efficient conditional diffusion and accelerated sampling, respectively, whereas SfmSR combines semantic guidance with few-step Flow Matching (Xiao et al., 2024; Meng et al., 2024; Gong et al., 2025). Beyond standard single-image settings, Ref-Diff introduces land-cover change priors for reference-based SR, NeurOp-Diff employs neural-operator diffusion for continuous-scale

restoration, and TexADiff addresses imbalanced remote-sensing textures through texture-aware diffusion (Dong et al., 2024; Xu et al., 2025; Zhang et al., 2026). Together, these studies show that conditioning design, transport parameterization, and texture modeling are active components of remote-sensing generative SR. Thus, semantic conditioning, residual transport, and few-step generation are not themselves our contributions. Our question is narrower: for a deterministic zero-residual source and a uniquely paired HR residual, does an explicit trajectory remain beneficial when the prescribed linear path has a constant analytical velocity target? Our conclusions are restricted to this formulation and do not extend to stochastic diffusion, noise-to-data Flow Matching, or multimodal generative settings.

Semantic and structural conditioning. Direct prediction still faces substantial ambiguity when high-frequency information is removed by downsampling. Prior work has therefore explored gradient-guided structure preservation and wavelet-based multi-scale restoration (Ma et al., 2020; Liu et al., 2018), while DINOv2 and FiLM provide powerful semantic representations and lightweight conditional modulation (Oquab et al., 2024; Perez et al., 2018). In DirectSR, these cues have distinct roles: frozen DINOv2 provides scene-level *what*, observation-derived differential responses provide pixel-aligned *where*, and LR consistency preserves *what has already been observed*. Together they constrain endpoint recovery rather than reduce it to unconstrained residual regression. Candidate-aware refinement is studied only as the optional DirectSR+ Stage 2, while DINO-FiLM semantic modulation and wavelet-based correction are examined as specific refinement variants.

3 METHOD

We propose **DirectSR**, a direct residual recovery framework for remote sensing super-resolution. It combines direct endpoint prediction with frozen multi-scale semantics, pixel-aligned structural cues, and LR consistency. DirectSR is the default reconstruction model, while **DirectSR+** is an optional refinement.

3.1 PROBLEM FORMULATION AND DIRECT RECOVERY

Let $y \in \mathbb{R}^{3\times h\times w}$ and $x \in \mathbb{R}^{3\times H\times W}$ denote the LR observation and HR target, where $H = sh$, $W = sw$, and $s \in \{2, 4\}$. Let U_s and D_s denote antialiased bicubic resizing operators, whose exact PyTorch implementation is specified in Appendix A.4. We define the bicubic reconstruction base and target residual as

$$B = U_s(y), \qquad R = x - B. \tag{1}$$

Here R denotes the complete correction relative to the bicubic base, rather than only high-frequency information.

Consider the zero-origin deterministic residual path

$$r_t = tR, \qquad t \sim \mathcal{U}(0, 1), \qquad v^* = \frac{\mathrm{d}r_t}{\mathrm{d}t} = R. \tag{2}$$

Although the analytical target is constant for each training pair, the input information changes along the path. For $t > 0$, the trajectory state directly reveals $R = r_t/t$, whereas this target-correlated information is unavailable at the zero-state inference start. The trajectory model is therefore trained on nonzero states that are not available at the initial inference point and may encounter off-trajectory states during numerical integration. Equation 2 does not establish that Flow Matching is redundant. Instead, it motivates a controlled comparison with an endpoint model that removes this trajectory-state dependence.

3.2 DIRECTSR: OBSERVATION-ALIGNED ENDPOINT PREDICTION

DirectSR fixes the residual state and time input to zero and predicts the endpoint residual directly:

$$\widehat{R} = f_\theta\left(0,\, B,\, \{F_l(B)\}_{l=1}^{3},\, S(B),\, 0\right). \tag{3}$$

The output represents the final HR residual rather than a velocity to be integrated. DirectSR is therefore not a trained Flow checkpoint evaluated for one step. The time-embedding layers are retained for the matched backbone comparison but always receive $t = 0$.

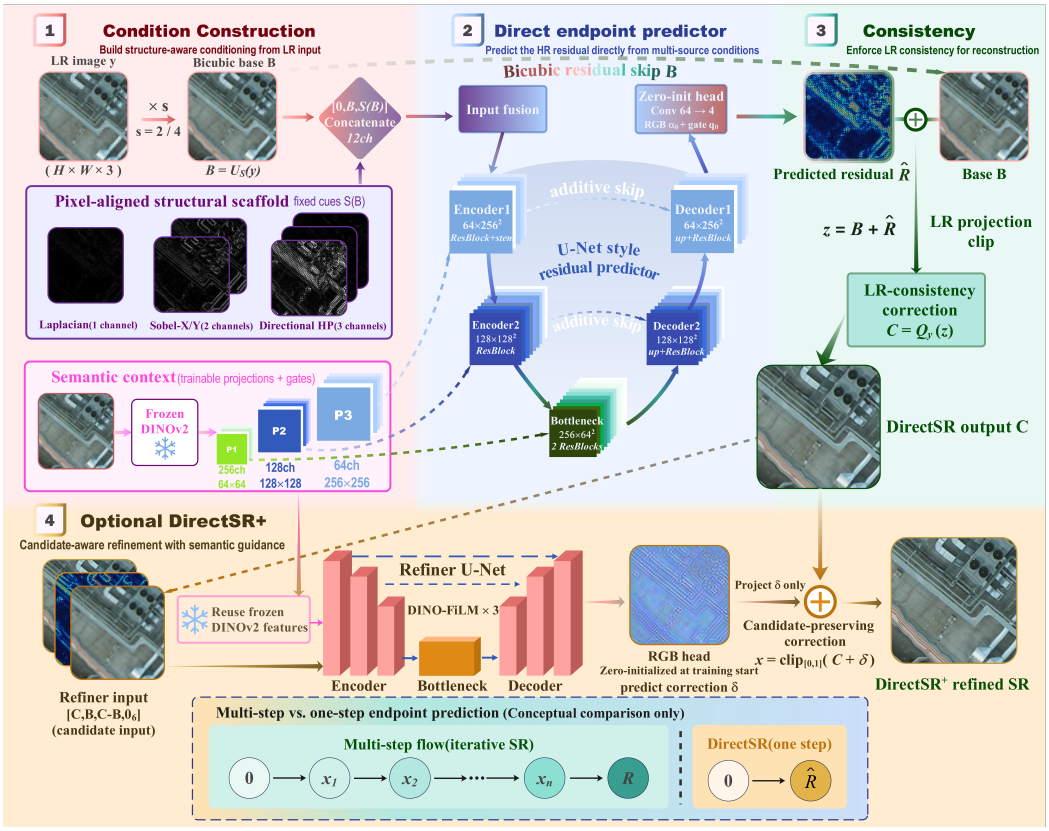


Figure 1: **Overview of DirectSR and optional DirectSR+ refinement.** **(1)** LR-derived semantic and structural conditions are constructed from the bicubic base. **(2)** DirectSR predicts the endpoint HR residual in one residual U-Net evaluation. **(3)** LR-consistency correction produces the DirectSR candidate. **(4)** DirectSR+ optionally applies candidate-aware refinement; the illustrated semantic-modulation variant uses DINO-FiLM. The bottom inset is a conceptual comparison between iterative residual Flow and direct endpoint prediction.

Figure 1 summarizes the framework. The LR observation provides the bicubic base, pixel-aligned differential cues, and frozen multi-scale DINOv2 features. After frozen DINOv2 feature extraction, DirectSR predicts the endpoint residual in one endpoint-predictor evaluation and applies LR-consistency correction to obtain the reconstruction. DirectSR+ optionally performs candidate-aware Stage 2 refinement; DINO-FiLM is evaluated as an additional semantic-modulation variant. The bottom inset conceptually contrasts multi-step residual Flow with direct endpoint prediction.

Semantic and structural conditioning. We use DINOv2 ViT-B/14 with register tokens (Oquab et al., 2024; Darcet et al., 2024). Frozen features from blocks $2, 5, 8$, and 11 form three semantic sources:

$$\begin{aligned} A_1 &= \Phi_2(B), \qquad A_2 = \Phi_5(B), \\ A_3 &= \mathrm{Concat}(\Phi_8(B), \Phi_{11}(B)), \\ F_l(B) &= \mathcal{R}_l[\sigma(g_l) P_l(A_l)], \qquad h_l \leftarrow h_l + F_l(B). \end{aligned} \tag{4}$$

The projections produce $64/128/256$-channel conditions for the three U-Net scales. DINO remains frozen, while P_l and g_l are trained jointly with the reconstruction network.

With grayscale conversion $G(u) = 0.299u_R + 0.587u_G + 0.114u_B$, we compute six fixed, pixel-aligned structural responses from the grayscale bicubic base:

$$S(B) = \big[S_x(G(B)), S_y(G(B)), L(G(B)), H_1(G(B)), H_2(G(B)), H_3(G(B))\big].$$

Here, S_x, S_y, L denote normalized Sobel-X, Sobel-Y, and Laplacian operators, while H_1, H_2, H_3 are normalized directional high-pass operators. All cues are derived exclusively from the LR-based

bicubic image. The two branches provide complementary semantic context about *what* may be present and pixel-aligned evidence about *where* local structures occur.

Residual prediction and observation consistency. The backbone is a three-scale U-Net-style residual encoder–decoder (Ronneberger et al., 2015), with $64/128/256$ channels and additive skip connections. Its 12-channel input and bounded residual output are

$$\begin{aligned} I_{\text{direct}} &= \text{Concat}\left(0_{3\times H\times W}, B, S(B)\right), \\ \widehat{R} &= 0.30\tanh(a_\theta) \odot [0.05 + 0.95\sigma(q_\theta)]. \end{aligned} \tag{5}$$

The output convolution is zero-initialized, so initially $\widehat{R} = 0$ and the uncorrected reconstruction equals B.

With $z = B + \widehat{R}$, the final candidate is

$$C = Q_y(z) = \text{clip}_{[0,1]}\left[z + U_s\big(y - D_s(z)\big)\right]. \tag{6}$$

Since $D_s U_s \neq I$ in general, Q_y is an approximate back-projection rather than an exact projection. It preserves content supported by the observed LR image.

3.3 Training DirectSR

Let

$$\rho(e) = \text{mean}\,\sqrt{e^2 + 10^{-6}},$$

and define the differential-response operator as

$$S_d(u) = \big[S_x(G(u)), S_y(G(u)), L(G(u))\big].$$

For $z = B + \widehat{R}$, we optimize

$$\begin{aligned} \mathcal{L}_{\text{DirectSR}} = 1.5\,\rho(z - x) &+ 0.15\,\|S_d(z) - S_d(x)\|_1 \\ &+ 0.20\,\|D_s(z) - y\|_1. \end{aligned} \tag{7}$$

Since $\widehat{R} - R = z - x$, the residual and image penalties coincide and are combined with weight 1.5. This preserves the matched loss weighting, but not the training-state distribution. DINO remains frozen, while all other DirectSR modules are trained. No adversarial or LPIPS loss is used.

3.4 Optional DirectSR+ Refinement

DirectSR Stage 1 is the complete default reconstruction model. **DirectSR+** is an optional Stage 2 refinement that examines whether the frozen Stage-1 candidate can benefit from an additional candidate-aware correction. During Stage 2 training, the complete DirectSR Stage 1 is frozen and only the candidate-aware RGB refiner is optimized; DINO-FiLM is evaluated as an optional semantic-modulation variant using separate trainable mappings over the frozen semantic features. The candidate-preserving correction guarantees that zero refinement returns the Stage-1 candidate exactly. DirectSR+ is therefore not a mandatory component of DirectSR, and its benefit is treated as dataset- and setting-dependent. Architecture and loss definitions are provided in Appendix A.2.

4 Experimental Results

We evaluate DirectSR in terms of reconstruction fidelity, structural fidelity, performance across separately trained scale settings, and mechanism validation. The $4\times$ setting serves as our primary benchmark, while $2\times$ SR further examines the reconstruction ceiling under milder degradation. Dedicated matched residual-Flow comparisons, multi-seed semantic ablations, and optional DirectSR+ refinement are then used to isolate the main design choices. Potsdam and Toronto constitute the primary benchmarks in the main paper, while additional results on AID are provided in Appendix B.2. Unless otherwise stated, "best" and "state-of-the-art (SOTA)" refer to the best results among the methods compared under our unified evaluation protocol.

Table 1: **Main reconstruction comparison under $4\times$ SR.** "Gen." indicates whether the method follows a generative SR formulation. The DirectSR row reports the complete seed-618 evaluation. Best and second-best results are shown in **bold** and underlined, respectively.

		Potsdam		Toronto	
Method	Gen.	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Bicubic	–	31.2381	0.810072	25.3381	0.746314
EDSR	–	**34.2472**	0.866276	28.7081	0.853507
RCAN	–	34.1592	0.865256	28.7126	0.853505
SwinIR	–	34.1666	0.865424	28.6682	0.852608
HAT	–	34.1177	0.864918	28.5749	0.850399
SfmSR (official, $\lambda = 0.1$)	✓	32.8259	0.834503	27.3962	0.812518
FastDiffSR	✓	33.1266	0.841475	27.9013	0.833281
ResShift	✓	31.2611	0.783465	–	–
SinSR	✓	29.3787	0.727963	–	–
DirectSR (Ours)	–	34.1942	**0.867306**	**29.1091**	**0.861453**

4.1 Experimental Protocol

Potsdam and Toronto use fixed paired test sets of $3{,}272$ and $1{,}276$ patches, respectively, while AID uses $1{,}000$ fixed 256×256 test images. The $2\times$ and $4\times$ settings correspond to $128 \to 256$ and $64 \to 256$, respectively. Toronto is used only for cross-region evaluation with the Potsdam-trained model and is excluded from training, validation, and checkpoint selection. AID LR inputs are generated online using the PyTorch bicubic operator defined in Appendix A.4, whereas Potsdam and Toronto use fixed aligned LR–HR pairs. All primary metrics are computed on saved uint8 RGB PNG outputs over the full image without border cropping. Dataset-partition audits found no exact cross-split RGB duplicates, with additional geospatial non-overlap verified for Potsdam.

We report PSNR/SSIM for reconstruction fidelity, LPIPS/DISTS/FID for perceptual quality, MUSIQ/CLIP-IQA+ as additional no-reference indicators, and Gradient-L1, HF-L1, directional Haar errors, and Edge PSNR for structural fidelity. **DirectSR** denotes the default Stage-1 endpoint model, whereas **DirectSR+** denotes optional candidate-aware Stage-2 refinement; the specific refinement variant is stated when relevant. For controlled comparison, the **Matched Residual Flow** uses the same backbone and observation conditions as DirectSR, with four integration steps by default. Protocol-compatible official checkpoints are used for generative baselines when available.

4.2 Main Results under $4\times$ SR

We regard $4\times$ SR as the primary setting because substantial boundary and high-frequency information is no longer directly observable from the LR input. Table 1 summarizes the main reconstruction results.

On Potsdam, DirectSR achieves 34.1942 dB PSNR and 0.867306 SSIM, remaining only 0.0530 dB below EDSR in PSNR while obtaining the highest SSIM among the compared methods. On the cross-region Toronto benchmark, DirectSR reaches 29.1091 dB / 0.861453, improving PSNR over RCAN by 0.3965 dB while again achieving the highest SSIM. Importantly, these results are obtained by the *default DirectSR Stage-1 endpoint model*, without optional Stage 2 refinement. Under our unified protocol, DirectSR therefore achieves the strongest reconstruction fidelity among the compared methods on Toronto $4\times$.

Figure 2 provides a representative visual comparison across deterministic, generative, and matched trajectory-based methods. The upper rows compare deterministic reconstruction methods, whereas the lower rows compare generative methods; the Matched 4-step Flow and the *default DirectSR Stage-1 output* are included in both groups for reference.

All reconstructions use identical enlarged regions, with error maps showing mean absolute RGB error to HR over a shared range of $[0, 0.15]$. DirectSR preserves roof boundaries and repeated line

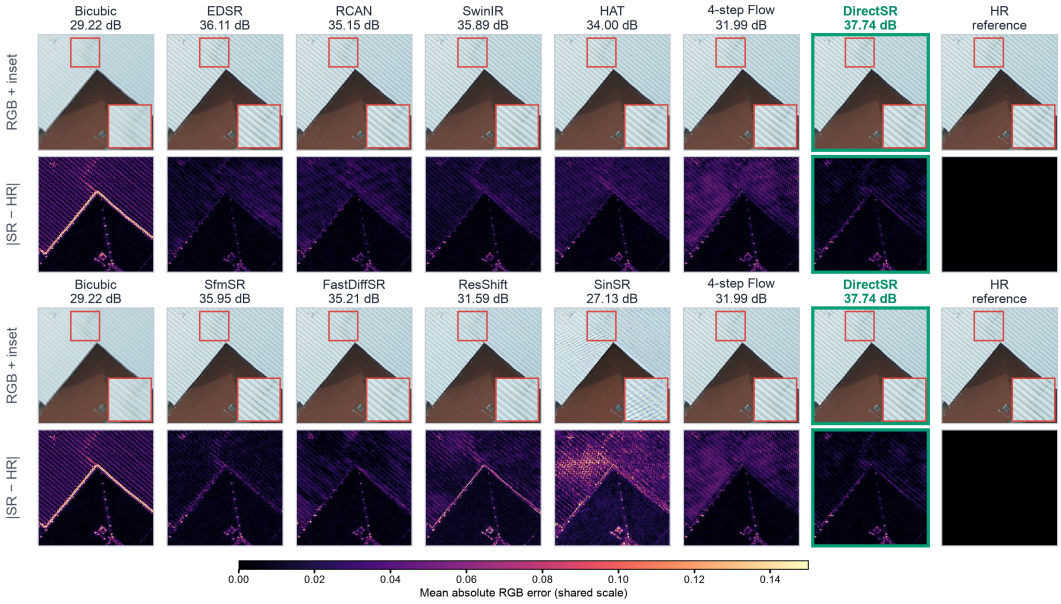


Figure 2: **Comprehensive qualitative comparison under $4\times$ SR.** The upper pair of rows compares Bicubic, EDSR, RCAN, SwinIR, HAT, the Matched 4-step Flow, DirectSR, and HR, while the lower pair compares Bicubic, SfmSR, FastDiffSR, ResShift, SinSR, the Matched 4-step Flow, DirectSR, and HR. RGB results include enlarged views of identical local regions. The corresponding $|\mathrm{SR} - \mathrm{HR}|$ maps use a shared mean-absolute-RGB-error range of $[0, 0.15]$, enabling direct comparison of spatial reconstruction errors. All DirectSR panels show outputs of the default Stage-1 endpoint predictor; optional Stage-2 refinement is not applied.

Table 2: **Structural fidelity comparison on Toronto $4\times$.** Lower values indicate better fidelity for Gradient-L1, HF-L1, and directional Haar errors, while higher Edge PSNR is better. Best and second-best results are shown in **bold** and underlined, respectively.

Method	Grad-L1↓	HF-L1↓	LH-L1↓	HL-L1↓	HH-L1↓	Edge PSNR↑
Bicubic	0.016688	0.019975	0.023698	0.025172	0.011057	28.6823
EDSR	0.012838	0.016332	<u>0.018803</u>	0.020143	0.010050	32.4175
RCAN	<u>0.012836</u>	<u>0.016329</u>	0.018806	<u>0.020139</u>	<u>0.010043</u>	<u>32.4363</u>
HAT	0.013030	0.016569	0.019098	0.020446	0.010162	32.3052
SfmSR	0.015126	0.018616	0.021814	0.022997	0.011038	31.3124
FastDiffSR	0.014042	0.017737	0.020660	0.021791	0.010761	32.2373
DirectSR (Ours)	**0.012367**	**0.015761**	**0.018151**	**0.019356**	**0.009777**	**32.9212**

structures more accurately with lower residual error. All DirectSR panels show the default Stage-1 output. Quantitative conclusions are reported in Tables 1 and 2.

The Toronto advantage is not limited to PSNR and SSIM. As shown in Table 2, DirectSR obtains the lowest Gradient-L1 (0.012367), HF-L1 (0.015761), and all three directional Haar errors, while achieving the highest Edge PSNR of 32.9212 dB. DirectSR ranks first on every structural metric reported in Table 2. Together with Table 1, these results show that DirectSR provides the strongest reconstruction and structural fidelity among the methods included in both Toronto evaluations.

Perceptual metrics exhibit a different operating point. Generative methods such as SfmSR, FastDiffSR, and ResShift obtain stronger LPIPS, DISTS, or FID in several settings, whereas DirectSR prioritizes correspondence with the paired HR target. This behavior is consistent with the perception–distortion trade-off. Complete perceptual results are reported in Appendix B.4.

Table 3: **Reconstruction fidelity under $2\times$ SR.** All values are evaluated from saved uint8 RGB outputs using the unified IQA protocol. DirectSR denotes the endpoint-only Stage-1 model. DirectSR+ reports the mean over three refinement seeds, with the standard deviation reported for PSNR. Best and second-best results are shown in **bold** and underlined, respectively.

	Potsdam		Toronto	
Method	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Bicubic	37.0078	0.944079	31.2305	0.927723
EDSR	<u>41.8525</u>	0.974285	<u>37.2704</u>	<u>0.970887</u>
RCAN	41.8348	0.974419	37.1780	0.970683
SwinIR	41.7388	0.974090	37.0561	0.970233
SfmSR (retrained)	40.8114	0.968443	36.1890	0.964584
Matched 4-step Flow + Refiner	41.6505	0.973808	36.8686	0.969637
DirectSR (Stage-1, Ours)	41.8328	<u>0.974474</u>	37.1296	0.970643
DirectSR+ (Ours)	$\mathbf{41.9220}$ ± 0.0029	**0.974869**	$\mathbf{37.3111}$ ± 0.0028	**0.971139**

4.3 Results under $2\times$ SR

The $2\times$ experiments further examine reconstruction under milder degradation. Under the unified evaluation protocol, DirectSR Stage-1 reaches 41.8328 dB / 0.974474 on Potsdam and 37.1296 dB / 0.970643 on Toronto. The optional DirectSR+ configuration further reaches three-seed mean PSNRs of 41.9220 ± 0.0029 dB and 37.3111 ± 0.0028 dB, respectively, while achieving the highest PSNR and SSIM among the compared methods on both datasets. Under our unified protocol, DirectSR+ therefore achieves **state-of-the-art reconstruction fidelity on both Potsdam and Toronto** $2\times$ among the compared methods. Together with the separately trained $4\times$ models, these results show that the DirectSR formulation remains effective across both scale settings, while optional refinement further raises the reconstruction ceiling under milder degradation.

To directly compare endpoint and trajectory-based recovery under matched conditions, we conduct a dedicated experiment with a **Matched 4-step Flow + Refiner** using the corresponding trajectory-based formulation.

We additionally evaluate AID as a third dataset to examine robustness across scene distributions rather than to make an additional SOTA claim. On AID $2\times$, DirectSR reaches 36.4093 ± 0.0015 dB, only 0.0286 dB below RCAN. On AID $4\times$, it obtains 30.3018 dB / 0.821338, remaining close to the strongest deterministic baselines while substantially exceeding the evaluated generative baselines in PSNR. Complete AID results are reported in Appendix B.2.

4.4 Mechanism Validation

To compare trajectory-based and endpoint recovery under matched conditions, we use the same backbone, observation conditions, data split, optimization budget, and loss family. The trajectory-based counterpart is evaluated with $1/2/4/8$ integration steps, whereas DirectSR is independently retrained with the endpoint objective. Thus, the comparison isolates the recovery parameterization rather than merely the number of sampling steps.

As shown in Table 4, the Matched 4-step Flow reaches 39.3886 dB on Potsdam and 33.8664 dB on Toronto, while DirectSR achieves 41.8328 and 37.1296 dB, yielding gains of $+2.4442$ and $+3.2632$ dB. Even against the best tested Flow-step setting, DirectSR retains gains of $+2.0892$ and $+3.0332$ dB, supporting direct endpoint recovery under the deterministic residual formulation studied here.

We further examine the contribution of Stage-1 semantic conditioning. Across three independent training seeds, DINO improves PSNR by 0.03557 dB on average on AID $2\times$ and by 0.13892 dB on Potsdam $4\times$. All six fixed-model paired-image bootstrap 95% confidence intervals lie above zero, supporting a consistent benefit from semantic conditioning in the tested settings. These intervals reflect image-level differences for fixed trained model pairs rather than uncertainty over training randomness. Because dataset and scale vary jointly, we do not interpret them as evidence of same-scale cross-dataset generality. Complete results are provided in Appendix B.6. The matched Flow

Table 4: **Matched comparison between residual Flow trajectories and direct endpoint recovery.** All values are PSNR (dB).

Method	Potsdam 2×	Toronto 2×
Matched Flow (1 step)	39.7436	34.0964
Matched Flow (2 steps)	39.6067	34.0204
Matched Flow (4 steps)	39.3886	33.8664
Matched Flow (8 steps)	39.4556	33.8383
DirectSR	**41.8328**	**37.1296**
Gain over best Flow	**+2.0892**	**+3.0332**

Table 5: **Model complexity and inference efficiency.** All measurements are obtained on a single RTX 3090 in FP32. "Updated params." denotes parameters optimized in the reported training stage. The DirectSR+ row corresponds to the DINO-FiLM refinement configuration. Peak memory denotes allocated GPU memory.

	Model size		Batch size 1		Batch size 16	
Method	Params.	Updated params.	Latency↓ (ms)	Peak mem.↓ (MiB)	Throughput↑ (img/s)	Peak mem.↓ (MiB)
DirectSR	91.851M	5.269M	**16.04**	**508**	**105.63**	**2969**
DirectSR+	94.341M	2.490M	21.60	521	70.01	2982

comparison is restricted to the deterministic zero-origin residual formulation studied here and should not be generalized to stochastic Flow Matching or diffusion models.

4.5 Optional Refinement and Efficiency

DirectSR+ is an optional candidate-aware refinement rather than a replacement for DirectSR Stage 1. On the final seed-618 Stage-1 candidate, candidate-aware refinement improves Toronto $4\times$ by 0.0571 dB, whereas DINO-FiLM yields only a 0.0039 dB gain. On Potsdam $4\times$, the two variants decrease PSNR by 0.0152 and 0.0201 dB, respectively, despite improving LPIPS and DISTS. These results reveal a clear dataset-dependent fidelity–perception trade-off. We therefore retain DirectSR Stage 1 as the default model and treat DirectSR+ as an optional refinement. Detailed results are provided in Appendix B.7.

Efficiency is measured on an NVIDIA RTX 3090 in FP32 for $64 \times 64 \rightarrow 256 \times 256$ $4\times$ SR. Each model is evaluated in an independent process using 10 warm-up and 50 measured iterations. DirectSR requires 16.04 ms per image at batch size 1 and reaches 105.63 images/s at batch size 16. The DINO-FiLM configuration of DirectSR+ requires 21.60 ms and reaches 70.01 images/s, respectively. Here, updated parameters refer to those optimized in the corresponding training stage; DirectSR+ contains 7.759M learned parameters across both stages, while Stage 2 updates only the additional 2.490M refiner parameters.

5 Conclusion

This work presents **DirectSR**, which reformulates remote sensing super-resolution as single-pass endpoint residual recovery. Starting from a bicubic observation anchor, DirectSR combines frozen multi-scale DINOv2 semantics, pixel-aligned differential responses, and LR-consistency correction to recover the HR residual without iterative diffusion sampling or ODE integration. Under our unified protocol, DirectSR achieves the strongest reconstruction and structural fidelity among the methods in the main Toronto $4\times$ comparison, while remaining competitive on Potsdam $4\times$; optional DirectSR+ further improves $2\times$ reconstruction. Matched residual-Flow comparisons and multi-seed semantic ablations support the effectiveness of direct endpoint recovery and Stage-1 semantic conditioning, respectively. Final-candidate experiments further show that Stage 2 refinement is dataset dependent, reinforcing DirectSR Stage 1 as the core default model and DirectSR+ as an optional, setting-dependent extension.

AI USE STATEMENT

We used generative AI tools to assist with research framing, feedback on methodology and experimental design, dataset preparation and reformatting, interpretation of experimental results, manuscript organization, drafting, translation, language editing, figure revision, and LaTeX typesetting. All experimental results were produced and verified by the authors. We reviewed all AI-assisted content and take full responsibility for the final manuscript.

REFERENCES

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6228–6237, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Blau_The_Perception-Distortion_Tradeoff_CVPR_2018_paper.html.

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*, pp. 22367–22377, 2023.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision – ECCV 2014*, volume 8692 of *Lecture Notes in Computer Science*, pp. 184–199. Springer, 2014. doi: 10.1007/978-3-319-10593-2_13. URL https://doi.org/10.1007/978-3-319-10593-2_13.

Runmin Dong, Shuai Yuan, Bin Luo, Mengxuan Chen, Jinxiao Zhang, Lixian Zhang, Weijia Li, Juepeng Zheng, and Haohuan Fu. Building bridges across spatial and temporal resolutions: Reference-based super-resolution via change priors and conditional diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27674–27684, June 2024. doi: 10.1109/CVPR52733.2024.02614.

Zhicheng Gong, Fangzhou Yi, Ling Guan, Chunzhu Dong, and Hui Zeng. Semantic-guided flow matching for fast and accurate remote sensing image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–12, 2025. doi: 10.1109/TGRS.2025.3608455. URL https://doi.org/10.1109/TGRS.2025.3608455.

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, June 2016. doi: 10.1109/CVPR.2016.182. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Kim_Accurate_Image_Super-Resolution_CVPR_2016_paper.html.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, June 2026. URL https://openaccess.thecvf.com/CVPR2026?day=2026-06-07.

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 Workshops*, pp. 1833–1844, 2021. URL https://openaccess.thecvf.com/content/ICCV2021W/AIM/papers/Liang_SwinIR_Image_Restoration_Using_Swin_Transformer_ICCVW_2021_paper.pdf.

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*, pp. 136–144, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017_workshops/w12/html/Lim_Enhanced_Deep_Residual_CVPR_2017_paper.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level Wavelet-CNN for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 773–782, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018_workshops/w13/html/Liu_Multi-Level_Wavelet-CNN_for_CVPR_2018_paper.html.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.03003.

Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. Structure-preserving super resolution with gradient guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7769–7778, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Ma_Structure-Preserving_Super_Resolution_With_Gradient_Guidance_CVPR_2020_paper.html.

Fanen Meng, Yijun Chen, Haoyu Jing, Laifu Zhang, Yiming Yan, Yingchao Ren, Sensen Wu, Tian Feng, Renyi Liu, and Zhenhong Du. A conditional diffusion model with fast sampling strategy for remote sensing image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–16, 2024. doi: 10.1109/TGRS.2024.3458009.

Maxime Oquab, Timothee Darcet, Theo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3942–3951, 2018. doi: 10.1609/aaai.v32i1.11671. URL https://ojs.aaai.org/index.php/AAAI/article/view/11671.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351 of *Lecture Notes in Computer Science*, pp. 234–241. Springer, Cham, 2015. doi: 10.1007/978-3-319-24574-4_28. URL https://doi.org/10.1007/978-3-319-24574-4_28.

Yufei Wang, Wenhan Yang, Xinyuan Chen, Yaohui Wang, Lanqing Guo, Lap-Pui Chau, Ziwei Liu, Yu Qiao, Alex C. Kot, and Bihan Wen. SinSR: Diffusion-based image super-resolution in a single step. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25796–25805, 2024.

Yi Xiao, Qiangqiang Yuan, Kui Jiang, Jiang He, Xianyu Jin, and Liangpei Zhang. EDiffSR: An efficient diffusion probabilistic model for remote sensing image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–14, 2024. doi: 10.1109/TGRS.2023.3341437.

Zihao Xu, Yuzhi Tang, Bowen Xu, and Qingquan Li. NeurOp-Diff: Continuous remote sensing image super-resolution via neural operator diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12491–12501, October 2025. doi: 10.1109/ICCV51701.2025.01161.

Zongsheng Yue, Jianyi Wang, and Chen Change Loy. ResShift: Efficient diffusion model for image super-resolution by residual shifting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 13294–13307, 2023. doi: 10.52202/075280-0583. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/2ac2eac5098dba08208807b65c5851cc-Abstract-Conference.html.

Enzhuo Zhang, Sijie Zhao, Dilxat Muhtar, Zhenshi Li, Xueliang Zhang, and Pengfeng Xiao. Remote sensing image super-resolution for imbalanced textures: A texture-aware diffusion framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38238–38247, 2026.

Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision*, pp. 286–301, 2018. URL https://openaccess.thecvf.com/content_ECCV_2018/html/Yulun_Zhang_Image_Super-Resolution_Using_ECCV_2018_paper.html.

A Additional Implementation Details

This section provides implementation details omitted from the main paper. We first describe the core DirectSR endpoint predictor, including semantic feature extraction, fixed differential conditions, backbone design, output parameterization, and observation correction. We then detail the optional DirectSR+ candidate-aware refiner. DirectSR constitutes the default recovery path, whereas DirectSR+ is an optional extension.

A.1 DirectSR Stage-1 Implementation

Multi-scale semantic conditions. DirectSR uses a frozen DINOv2 ViT-B/14 encoder with register tokens. The bicubic base B is resized to 252×252 before being passed to DINOv2. Let Z_k denote the token representation from zero-indexed block k. We construct

$$A_1 = Z_2, \qquad A_2 = Z_5, \qquad A_3 = \mathrm{Concat}(Z_8, Z_{11}). \tag{8}$$

The three sources are projected to 64, 128, and 256 channels, gated, and then resized to the corresponding backbone grids:

$$F_l = \mathcal{R}_l[\sigma(g_l) \odot P_l(A_l)], \qquad h_l \leftarrow h_l + F_l. \tag{9}$$

Here, P_l is the trainable 1×1 projection, g_l is the learned gate, and $\mathcal{R}_l$ denotes spatial alignment to the receiving backbone resolution. The DINOv2 encoder remains frozen, while the projections and gates are trained jointly with the Stage-1 endpoint predictor.

Observation-derived differential responses. In addition to semantic context, DirectSR constructs six fixed, pixel-aligned differential responses from the bicubic observation anchor. We first convert an RGB image u to grayscale as

$$G(u) = 0.299u_R + 0.587u_G + 0.114u_B, \tag{10}$$

and define

$$S(u) = \big[S_x(G(u)), S_y(G(u)), L(G(u)), H_1(G(u)), H_2(G(u)), H_3(G(u))\big], \tag{11}$$

where S_x, S_y, and L denote normalized Sobel-X, Sobel-Y, and Laplacian operators, and H_1, H_2, H_3 are normalized directional high-pass operators. Their normalization factors are $8, 8, 4, 6, 6, 6$, respectively.

All responses are derived exclusively from the LR-based bicubic observation and contain no learned parameters or HR-derived edge labels. They therefore provide pixel-aligned structural cues complementary to the DINO semantic context.

Endpoint predictor input and backbone. DirectSR does not start from random noise or a nonzero trajectory state. In direct mode, the residual-state input is fixed to a three-channel zero tensor and the time input is fixed to $t = 0$. The complete network input is therefore

$$I_{\mathrm{Direct}} = \mathrm{Concat}\left[\mathbf{0}_{3\times H\times W}, B, S(B)\right] \in \mathbb{R}^{12\times H\times W}. \tag{12}$$

The endpoint predictor is a three-scale residual encoder–decoder with feature widths 64, 128, and 256 and additive skip connections. The semantic conditions F_1, F_2, F_3 are additively injected into the corresponding feature scales. The time-embedding pathway is retained to preserve architectural comparability with the matched residual-Flow counterpart, but DirectSR always evaluates it at $t = 0$.

Residual amplitude and spatial gate. The output head predicts four channels: three RGB residual-amplitude channels a_θ and one spatial gate q_θ shared across the RGB channels. The gate modulates the residual magnitude and should not be interpreted as an uncertainty estimate.

The predicted residual is

$$\widehat{R} = 0.30\tanh(a_\theta) \odot [0.05 + 0.95\sigma(q_\theta)]. \tag{13}$$

The pre-correction reconstruction is

$$z = B + \widehat{R}. \tag{14}$$

The final output convolution is zero-initialized. Consequently, $\widehat{R} = 0$ at initialization and the initial reconstruction is exactly the bicubic anchor $z = B$.

Observation-consistency correction. After endpoint prediction, DirectSR applies an LR-consistency correction:

$$Q_y(z) = \mathrm{clip}_{[0,1]}\left[z + U_s\big(y - D_s(z)\big)\right], \tag{15}$$

where D_s and U_s denote the degradation/downsampling operator and its corresponding upsampling operator.

Because $D_s U_s \neq I$ in general, this operation is an approximate back-projection or observation-consistency correction rather than an exact orthogonal projection. The final DirectSR candidate is

$$C = Q_y(z). \tag{16}$$

DirectSR training objective. We use the Charbonnier function

$$\rho(e) = \mathrm{mean}\,\sqrt{e^2 + 10^{-6}}. \tag{17}$$

We define the grayscale differential-response operator as

$$S_d(u) = \left[S_x(G(u)), S_y(G(u)), L(G(u))\right], \tag{18}$$

where $G(\cdot)$ denotes grayscale conversion, and S_x, S_y, and L are the fixed Sobel-X, Sobel-Y, and Laplacian operators.

The DirectSR objective is

$$\mathcal{L}_{\mathrm{Direct}} = 1.5\,\rho(z - x) + 0.15\,\|S_d(z) - S_d(x)\|_1 + 0.20\,\|D_s(z) - y\|_1. \tag{19}$$

All Stage-1 training losses are evaluated on z, before the final observation-consistency correction and clipping. The default Stage-1 model uses neither adversarial nor LPIPS supervision and contains no independent learned structure-prediction head.

For the matched direct formulation,

$$\widehat{R} - R = z - x, \tag{20}$$

so residual prediction and image reconstruction errors coincide numerically. The reconstruction coefficient 1.5 preserves the corresponding loss family and weighting used in the matched comparison; it does not imply that DirectSR and residual Flow share the same training-state distribution.

Single-pass inference. At inference time, DINOv2 performs one frozen feature-extraction pass, followed by a single endpoint-predictor evaluation and one observation-consistency correction. No trajectory initialization, iterative solver, diffusion sampling, or ODE integration is required.

A.2 DirectSR+ Refinement

This section provides the implementation details of the optional DirectSR+ refiner introduced in Sec. 3.4. DirectSR+ is not part of the default DirectSR recovery path.

During refinement training, the entire trained Stage 1 is frozen, including the frozen DINOv2 encoder, the trained semantic projection/gating modules, and the endpoint predictor. The refiner is a separately trained candidate-aware encoder–decoder with feature widths 48, 96, and 192, and predicts only a three-channel RGB correction. Unless otherwise specified, the default configuration contains no learned structure-prediction or wavelet-output branch.

Candidate-aware input. Rather than using the DirectSR candidate alone, the refiner explicitly retains the candidate, the bicubic observation anchor, their difference, and the observation-derived differential cues. Let

$$\bar{B} = \mathrm{clip}_{[0,1]}(B). \tag{21}$$

Using the same six-channel grayscale structural scaffold $S(\bar{B})$, the refiner input is

$$I_+ = \mathrm{Concat}\left[C, \bar{B}, C - \bar{B}, S(\bar{B})\right] \in \mathbb{R}^{15\times H\times W}. \tag{22}$$

The 15 channels consist of 3 candidate channels, 3 bicubic channels, 3 candidate-relative residual channels, and 6 grayscale structural-response channels.

DINO-FiLM semantic modulation. The candidate-aware RGB refiner does not require semantic modulation. When the DINO-FiLM variant is enabled, the frozen Stage-1 semantic conditions F_l are reused to modulate the three refiner feature scales through FiLM. Let $\mathcal{R}_l$ denote spatial alignment to the corresponding refiner resolution. We apply

$$\widetilde{h}_l = h_l \odot [1 + 0.1 \tanh(\gamma_l(\mathcal{R}_l(F_l)))] + \beta_l(\mathcal{R}_l(F_l)). \tag{23}$$

Here, γ_l and β_l predict multiplicative and additive modulation parameters, respectively. The factor 0.1 bounds the multiplicative modulation around the identity transformation. Both FiLM mappings are zero-initialized, so semantic modulation initially leaves the refiner features unchanged.

Bounded RGB correction. The decoder predicts only a candidate-relative RGB correction:

$$\delta = 0.10 \tanh(H_{\mathrm{rgb}}(h_{\mathrm{dec}})), \tag{24}$$

where the RGB output head H_{rgb} is zero-initialized. The bounded parameterization limits the correction magnitude and gives $\delta = 0$ at initialization.

Candidate-preserving observation correction. A second full back-projection of $C + \delta$ could modify an already valid DirectSR candidate even when $\delta = 0$. We therefore apply observation correction only to the change introduced by the refinement. Define the unclipped correction operator

$$P_y(z) = z + U_s\big(y - D_s(z)\big). \tag{25}$$

The final refined output is

$$\widehat{x}_+ = \mathrm{clip}_{[0,1]}\left[C + P_y(C + \delta) - P_y(C)\right]. \tag{26}$$

When $\delta = 0$,

$$P_y(C + \delta) - P_y(C) = 0, \tag{27}$$

and therefore

$$\widehat{x}_+ = C. \tag{28}$$

DirectSR+ thus learns only a candidate-relative correction rather than reconstructing the complete image again.

Refinement objective. Using the same Charbonnier function

$$\rho(e) = \mathrm{mean}\sqrt{e^2 + 10^{-6}},$$

the refinement objective is

$$\mathcal{L}_{\mathrm{DirectSR+}} = \rho(\widehat{x}_+ - x) + 0.20\mathcal{L}^+_{\mathrm{diff}} + 0.05\mathcal{L}^+_{\mathrm{freq}} + 0.20\|D_s(\widehat{x}_+) - y\|_1. \tag{29}$$

The differential term reuses the grayscale response operator S_d defined for DirectSR:

$$\mathcal{L}^+_{\mathrm{diff}} = \|S_d(\widehat{x}_+) - S_d(x)\|_1. \tag{30}$$

The frequency term compares the log-magnitude spectra:

$$\mathcal{L}^+_{\mathrm{freq}} = \|\log(1 + |\mathcal{F}(\widehat{x}_+)|) - \log(1 + |\mathcal{F}(x)|)\|_1, \tag{31}$$

where $\mathcal{F}$ denotes the orthonormal real FFT. This term is used only as a training objective and introduces no additional learned frequency-prediction branch.

The candidate-aware Stage-2 refiner uses no adversarial, LPIPS, learned structure-head, or wavelet-head loss.

Ablation-only wavelet branch. For ablation purposes, we additionally evaluate a candidate-relative wavelet branch. This branch is not part of the candidate-aware base refiner and is enabled only in the corresponding ablation.

The wavelet head predicts only the candidate-relative directional detail coefficients

$$W_d(x) - W_d(C), \tag{32}$$

rather than reconstructing the complete wavelet representation. Its correction is synthesized from the predicted detail coefficients with a zero LL component.

As shown in the candidate-controlled ablations, the wavelet branch does not provide consistent improvements across datasets and is therefore disabled by default.

Table 6: **Frozen and trainable components.** "Train" denotes parameters updated in the corresponding stage.

Component	Stage-1	Stage-2
DINOv2 encoder	Frozen	Frozen
Semantic projections / gates	Train	Frozen
Time embedding	Train	Frozen
Endpoint encoder–decoder	Train	Frozen
Residual amplitude / gate head	Train	Frozen
DirectSR+ encoder–decoder	–	Train
DINO-FiLM mappings	–	Train if enabled
RGB correction head	–	Train
Ablation-only wavelet heads	–	Train if enabled

A.3 Frozen and Trainable Components

To clarify the optimization scope of the two stages, Table 6 summarizes the parameter status of each module. DINOv2 remains frozen throughout Stage-1 training, while the semantic projection/gating modules and endpoint predictor are optimized. During Stage-2 training, the entire trained DirectSR model is frozen and only the DirectSR+ refiner is optimized.

A.4 Implementation of the Resizing Operators

Both U_s and D_s are implemented using `torch.nn.functional.interpolate`, rather than MATLAB-style bicubic interpolation. Throughout the experiments, we use bicubic interpolation with `align_corners=False` and `antialias=True`.

Let $\mathcal{I}_{\mathrm{bic}}(\cdot\,;\cdot)$ denote this fixed PyTorch bicubic interpolation operator. The upsampling and downsampling operators are then defined as

$$\begin{aligned} U_s(y) &= \mathcal{I}_{\mathrm{bic}}\big(y;\ \texttt{scale_factor} = s\big), \\ D_s(z) &= \mathcal{I}_{\mathrm{bic}}\big(z;\ \texttt{size} = (h, w)\big). \end{aligned} \tag{33}$$

Thus, U_s upsamples an LR image by the integer scale factor s, whereas D_s resizes an HR-space tensor to the explicit spatial size of the corresponding LR observation. No explicit padding is introduced by our implementation; boundary handling follows the behavior of PyTorch bicubic interpolation.

The same resizing definition is used when constructing the bicubic base $B = U_s(y)$, evaluating the LR reconstruction term $\|D_s(z) - y\|_1$, and applying the observation-consistency correction Q_y.

The LR-generation protocol differs between AID and the paired Potsdam/Toronto datasets. For the synthetic AID protocol, the LR observation is generated online from the augmented HR crop using the same operator D_s, followed by clipping to $[0, 1]$. For Potsdam and Toronto, the LR images are instead loaded from fixed, pre-generated, spatially aligned LR–HR pairs. Their LR test inputs are therefore not regenerated as $D_s(x)$ during evaluation; for these paired datasets, D_s is used in the LR reconstruction loss and the approximate observation-consistency correction.

A.5 Training Hyperparameters and Checkpoint Selection

Optimization. DirectSR and DirectSR+ are optimized sequentially rather than jointly. During Stage 1, only the pretrained DINOv2 encoder is frozen. DINOv2 is kept in evaluation mode and executed under `torch.no_grad()`. The trainable Stage 1 components comprise the multi-scale semantic projections, semantic gates, time embedding, endpoint encoder–decoder, and residual amplitude/gate head.

The Sobel, Laplacian, and three directional high-pass operators used to construct the pixel-aligned differential cues are fixed analytical filters and contain no trainable parameters.

Stage 2 is trained only after Stage 1 optimization is complete. The entire optimized Stage 1 model is then frozen, including DINOv2, the Stage 1 semantic projections and gates, the time embedding, the endpoint encoder–decoder, and the residual amplitude/gate head. Consequently, Stage 2 optimization cannot modify either the Stage 1 semantic representation or the endpoint candidate generator.

Table 7: **Training hyperparameters of DirectSR and DirectSR+.** Batch sizes are reported per GPU and globally. All models operate on 256×256 HR patches.

Setting	DirectSR Stage 1	DirectSR+ Stage 2
Optimizer	AdamW, $\beta = (0.9, 0.99)$, weight decay 10^{-4}	AdamW, $\beta = (0.9, 0.99)$, weight decay 10^{-4}
Maximum updates	100k	100k
Initial learning rate	2×10^{-4}	1×10^{-4}
Warmup	500 updates	250 updates
Learning-rate schedule	Linear warmup + cosine decay to zero	Linear warmup + cosine decay to zero
Gradient clipping	Global ℓ_2 norm 1.0	Global ℓ_2 norm 1.0
Precision	BF16 forward; FP32 losses and metric-sensitive operations	BF16 forward; FP32 losses and metric-sensitive operations
Batch configuration	2/GPU on 4 GPUs; global batch 8	4/GPU on 4 GPUs for the final paired $4\times$ refinement suite; global batch 16. 2/GPU on 4 GPUs for paired $2\times$ and AID $4\times$; global batch 8
EMA / gradient accumulation	None / None	None / None
Validation interval	Every 5k updates	Every 5k updates
Checkpoint interval	Every 5k updates	Every 5k updates
Checkpoint selection	Highest validation RGB PSNR	Highest validation RGB PSNR
Default optimization seed	618	618
Additional optimization seeds	2025 and 3407	2025 and 3407 where reported

The candidate-aware RGB refiner and its zero-initialized RGB correction head are optimized during Stage 2. When DINO-FiLM is enabled, its separate three-scale FiLM mappings are optimized jointly with the refiner. The wavelet head is trainable only in the corresponding ablations.

Both stages use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and weight decay 10^{-4}. We use neither exponential moving averages nor gradient accumulation.

Stage 1 is trained for at most 100k optimizer updates with an initial learning rate of 2×10^{-4}. Stage 2 uses the same maximum update budget with an initial learning rate of 1×10^{-4}.

For initial learning rate η_0, warmup length w, total update budget K, and update index k, both stages use linear warmup followed by cosine decay:

$$\eta_k = \eta_0 \min\left(1, \frac{k}{w}\right) \frac{1 + \cos\left(\pi \frac{\max(k-w,0)}{\max(K-w,1)}\right)}{2}. \tag{34}$$

We use $w = 500$ for Stage 1 and $w = 250$ for Stage 2. Gradients are clipped to a global ℓ_2 norm of 1.0.

Network forward passes use BF16 autocasting, whereas losses and metric-sensitive operations are evaluated in FP32. Training terminates immediately if a non-finite loss or gradient norm is detected.

The optimization scope of DirectSR and DirectSR+ is summarized in Table 6. The remaining training hyperparameters are reported below.

Training patches and augmentation. The HR training resolution is 256×256, corresponding to 128×128 LR inputs for $2\times$ SR and 64×64 LR inputs for $4\times$ SR.

For AID, random 256×256 HR crops are sampled from the original images before antialiased bicubic degradation. For the paired Potsdam-domain data, aligned LR–HR patches are already stored at the target spatial resolutions and are used directly.

Training augmentation consists of independent horizontal and vertical flips with probability 0.5, together with a rotation uniformly sampled from $\{0°, 90°, 180°, 270°\}$. We use no color jitter, MixUp, CutMix, synthetic noise, or additional sharpening.

AID validation uses deterministic center crops, whereas paired-domain validation uses fixed aligned patches. No augmentation is applied during validation.

Stage 2 candidate construction. For the separate multi-branch paired ablation suite, frozen Stage 1 candidates are generated once, quantized to 8-bit RGB, and stored using lossless PNG compression. For the final seed-618 $4\times$ refinement suite, Stage 1 candidates are generated once and

serialized as FP16 tensor files. Candidate-aware and DINO-FiLM refiners within this suite use the same fixed FP16 cache.

For AID, candidates are generated online using the frozen Stage 1 checkpoint. Candidate, bicubic, and HR tensors always receive identical geometric augmentation to preserve pixel-wise alignment.

Toronto is used exclusively for cross-region testing and is never used for training, validation, or checkpoint selection.

Data splits and random seeds. AID uses a category-wise $80/10/10$ train/validation/test split with split seed 618. The split remains fixed when the optimization seed is changed to 2025 or 3407, so the multi-seed experiments change optimization randomness without changing the underlying dataset partition.

For the paired Potsdam-domain experiments, validation filenames are fixed before training and shared by all variants. Each checkpoint evaluation uses at most 64 validation mini-batches.

Test-set metrics are never used to select checkpoint steps or tune optimization hyperparameters.

Checkpoint selection. All runs retain the checkpoint with the highest validation RGB PSNR rather than automatically using the final 100k checkpoint.

For the paired $4\times$ Stage 1 runs, the selected updates for seeds $618/2025/3407$ are 100k/100k/95k, respectively. For the final seed-618 paired $4\times$ refinement suite, the candidate-aware and DINO-FiLM refiners select their 55k and 5k checkpoints, respectively, according to validation RGB PSNR.

For paired $2\times$ DirectSR+, the selected Stage 2 checkpoints are 85k, 100k, and 100k for seeds 618, 2025, and 3407, respectively.

The AID $4\times$ Stage 1 run selects the 100k checkpoint. Its candidate-aware, DINO-FiLM, Wavelet, and DINO-FiLM+Wavelet refiners all select their 5k checkpoints according to validation RGB PSNR.

Checkpoint selection is based exclusively on the validation set. After the checkpoints are fixed, evaluation on the complete AID test set shows that all Stage 2 variants remain slightly below the frozen Stage 1 candidate. We therefore report Stage 1 DirectSR as the default AID output and use the refinement results to characterize the dataset-dependent behavior of Stage 2.

Software and hardware. Training uses Python 3.10.20, PyTorch 2.4.0 with CUDA 12.4, torchvision 0.19.0, and timm 1.0.27. All experiments are trained on NVIDIA GeForce RTX 3090 GPUs with 24 GB of memory.

Unless explicitly stated otherwise, all reported DirectSR and DirectSR+ results use the corresponding training configuration, validation protocol, and checkpoint-selection rule described above.

A.6 Implementation Conventions

Several implementation conventions are worth clarifying.

First, although the residual-state and time inputs of DirectSR are fixed to zero, the bicubic anchor, differential responses, and semantic conditions remain active; DirectSR is therefore not an unconditional zero-input predictor. Second, all Stage-1 training losses are evaluated on the pre-correction image z, whereas the inference candidate is $C = Q_y(z)$. Third, DirectSR+ always starts from the frozen Stage-1 candidate C, and its candidate-preserving correction guarantees exact recovery of C when the predicted refinement is zero.

Finally, the semantic features used by DirectSR+ are inherited from the frozen Stage-1 semantic pathway; DINOv2 is never fine-tuned during Stage-2 training.

B Additional Experimental Results

This section provides additional experimental results omitted from the main paper for space. We first report the dataset-partition audit and additional evaluation on AID, followed by detailed $2\times$ recon-

Table 8: **Reconstruction comparison on AID $2\times$.** DirectSR reports the three-seed mean. Best and second-best results are shown in **bold** and underlined, respectively.

Method	PSNR↑	SSIM↑
Bicubic	33.17632	0.907229
EDSR	36.43126	0.950364
RCAN	**36.43786**	**0.950506**
SwinIR	36.35065	0.949378
SfmSR	35.50318	0.941061
Matched 4-step Flow + Refiner	36.30430	0.949070
DirectSR	36.40931±0.00145	0.949952

struction and perceptual metrics, perceptual comparisons on Potsdam and Toronto $4\times$, multi-seed stability, semantic-conditioning analysis, Stage 2 refinement, and matched full-pipeline comparisons.

We additionally report matched trajectory-based full-pipeline comparisons to complement the Stage-1 formulation analysis in the main paper. Because the complete $1/2/4/8$-step Flow comparison is already reported in Sec. 4.4, it is not duplicated here.

B.1 Dataset Partition Audit

We audited all dataset partitions using decoded-RGB hashes and, where available, geospatial metadata.

For Potsdam, the training and test sets contain $12{,}321$ and $3{,}272$ georeferenced patches, respectively. We found no decoded-RGB duplicates across the two partitions and no positive-area spatial overlap between any training and test patch. Thus, the Potsdam split is disjoint at both the pixel and available geospatial levels.

The Toronto dataset contains $5{,}983$ images in its nominal training partition and $1{,}276$ images in the test partition. The nominal Toronto training partition is not used to train or select the DirectSR model evaluated in our cross-region experiments. No exact decoded-RGB duplicate was found between the two Toronto partitions. However, the released files do not retain sufficient geospatial coordinates or source-scene identifiers to verify geographic independence.

For AID, we use a deterministic category-wise $80/10/10$ split with split seed 618, resulting in $8{,}000/1{,}000/1{,}000$ train/validation/test images. The three partitions have no file-path intersections and no decoded-RGB duplicates across partitions. One duplicate pair exists within the training partition itself, but it does not cross a split boundary and therefore does not constitute train–validation or train–test leakage. AID does not provide sufficient geospatial or source-scene provenance for a geographic-overlap audit.

Accordingly, we claim verified pixel-level split disjointness for all three datasets and additional geographic non-overlap only for Potsdam.

B.2 Evaluation on the Additional AID Benchmark

AID provides a third remote-sensing scene distribution on which DirectSR is trained and evaluated independently. This experiment examines whether the formulation remains competitive across datasets and separately trained scale settings rather than testing zero-shot domain transfer. Under $2\times$ SR, DirectSR reaches a three-seed mean of 36.4093 ± 0.0015 dB, only 0.0286 dB below RCAN. Under $4\times$ SR, DirectSR reaches 30.3018 dB PSNR and 0.821338 SSIM, remaining close to the strongest deterministic baselines while substantially exceeding the evaluated generative methods in PSNR.

Perceptual metrics exhibit a different operating point, indicating that the principal advantage of DirectSR lies in paired reconstruction fidelity rather than uniformly minimizing perceptual distances.

These results show that the core endpoint predictor remains competitive on a third scene distribution and across both scale factors. At the same time, the perceptual metrics further demonstrate that DirectSR does not attempt to uniformly optimize perception-oriented criteria.

Table 9: **Additional benchmark comparison on AID $4\times$.** The DirectSR row reports the complete seed-618 evaluation. Best and second-best perceptual results are shown in **bold** and underlined, respectively.

Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓
Bicubic	28.09753	0.733990	0.455800	0.245010	99.9924
EDSR	**30.38230**	**0.823100**	0.279500	0.194300	85.4500
RCAN	30.37340	**0.823100**	0.279500	0.195200	85.8200
SfmSR (official)	28.77903	0.773180	0.206526	0.181006	74.7531
FastDiffSR	28.91333	0.779206	0.233982	0.175453	67.5445
ResShift	27.86980	0.736000	**0.188800**	**0.156200**	**54.2900**
SinSR	26.07670	0.677500	0.247400	0.200000	93.6400
DirectSR	30.30181	0.821338	0.280806	0.194229	84.3711

Table 10: **Detailed $2\times$ reconstruction and perceptual metrics.** All results are computed using the same unified evaluator.

Dataset	Output	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓
Potsdam $2\times$	Bicubic	37.00782	0.944079	0.130342	0.109298	14.3771
	DirectSR Stage-1	41.83275	0.974474	0.054536	0.052624	7.5701
	DirectSR+ seed618	41.92427	0.974871	0.052080	0.051607	7.3949
Toronto $2\times$	Bicubic	31.23053	0.927723	0.124997	0.117527	32.7756
	DirectSR Stage-1	37.12960	0.970643	0.042463	0.062109	16.1535
	DirectSR+ seed618	37.31145	0.971140	0.040573	0.061315	15.7815

Table 11: **Detailed perceptual comparison on Potsdam $4\times$.** Best and second-best results are shown in **bold** and underlined, respectively.

Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓
Bicubic	31.23811	0.810072	0.418196	0.238585	70.3300
EDSR	**34.24720**	0.866276	0.266133	0.170581	53.2071
RCAN	34.15923	0.865256	0.268841	0.171863	53.0511
SfmSR	32.82589	0.834503	**0.128958**	0.135720	36.3394
FastDiffSR	33.12655	0.841475	0.184649	**0.135238**	**24.3244**
ResShift	31.26113	0.783465	0.167544	0.145103	33.0716
SinSR	29.37867	0.727963	0.233894	0.182932	42.8361
DirectSR	34.19417	**0.867306**	0.266076	0.170299	52.4978

B.3 Detailed Metrics under $2\times$ SR

We additionally report detailed $2\times$ metrics produced by the same unified IQA evaluator. All values are computed from saved uint8 RGB outputs over the full image without border cropping.

B.4 Detailed Perceptual Results under $4\times$ SR

The main paper emphasizes reconstruction and structural fidelity. Here we provide detailed LPIPS, DISTS, and FID comparisons on Potsdam and Toronto $4\times$.

These results further illustrate that deterministic reconstruction and generative restoration methods occupy different operating points in the perception–distortion space. DirectSR prioritizes spatial correspondence with the paired HR target, whereas several generative approaches achieve stronger LPIPS, DISTS, or FID.

On Toronto, the principal advantage of DirectSR lies in reconstruction and structural fidelity rather than minimum perceptual distance. Accordingly, the SOTA claim in the main paper is restricted to reconstruction and structural fidelity and does not imply uniform superiority in LPIPS, DISTS, or FID.

B.5 Toronto $4\times$ Multi-Seed Stability

To verify that the Toronto $4\times$ result is not caused by a single training seed, we additionally evaluate three independently trained Stage-1 models.

Table 12: **Detailed perceptual comparison on Toronto $4\times$.** Best and second-best results are shown in **bold** and underlined, respectively.

Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓
Bicubic	25.33807	0.746314	0.436659	0.249354	118.4052
EDSR	28.70811	0.853507	0.213524	0.171311	88.1399
RCAN	28.71262	0.853505	0.217172	0.171944	86.0443
SwinIR	28.66818	0.852608	0.214496	0.172026	87.8377
HAT	28.57493	0.850399	0.224432	0.172852	82.7697
SfmSR	27.39617	0.812518	0.135684	0.155262	54.2178
FastDiffSR	27.90125	0.833281	**0.133370**	**0.133401**	**45.6798**
DirectSR	**29.10912**	**0.861453**	0.196211	0.164305	88.0307

Table 13: **Multi-seed stability on Toronto $4\times$.** All runs use validation-selected checkpoints.

Seed	Ckpt.	PSNR↑	Grad-L1↓	HF-L1↓	Edge PSNR↑
618	100k	29.10912	0.012367	0.015761	32.92117
2025	100k	28.89100	0.012622	0.016037	32.63864
3407	95k	28.90004	0.012611	0.016024	32.64840
Mean $\pm$ std.	–	28.96672$\pm$0.12341	0.012533$\pm$0.000144	0.015941$\pm$0.000156	32.73607$\pm$0.16038

Table 14: **Multi-seed analysis of DINO semantic conditioning.** The reported intervals are paired-image bootstrap 95% confidence intervals computed for each fixed trained model pair. PSNR values are taken from the paired-image evaluation underlying the bootstrap analysis; the main comparison tables report the corresponding unified aggregate IQA results.

Dataset	Seed	Full DINO	LR-only	ΔPSNR↑	95% CI
AID $2\times$	618	36.41090	36.36661	+0.04428	[0.03757, 0.05134]
	3407	36.40898	36.38043	+0.02855	[0.02023, 0.03664]
	2025	36.40806	36.37419	+0.03388	[0.02639, 0.04128]
Potsdam $4\times$	618	34.19429	33.99550	+0.19878	[0.18904, 0.20901]
	3407	34.09511	33.99236	+0.10275	[0.09706, 0.10876]
	2025	34.09235	33.97714	+0.11521	[0.10305, 0.12891]

Their PSNR values are 29.1091, 28.8910, and 28.9000 dB, respectively, all above RCAN (28.7126 dB) in the main comparison. Only metrics produced by the same evaluator are aggregated across seeds. The complete SSIM, LPIPS, and other metrics in the main paper remain the unified seed-618 evaluation and are not mixed with outputs from a different evaluation implementation.

B.6 Additional Analysis of Semantic Conditioning

The complete $1/2/4/8$-step matched Flow comparison is already reported in Sec. 4.4. We therefore avoid duplicating the Flow results and instead provide additional multi-seed analysis of semantic conditioning.

For each seed, the Full-DINO and LR-only configurations form a fixed trained pair. Paired bootstrap intervals are computed from per-image PSNR differences over the same test set. Accordingly, these intervals quantify paired differences over the test-image population rather than uncertainty arising from training randomness.

Averaged over three training seeds, DINO improves PSNR by 0.03557 dB on AID $2\times$ and 0.13892 dB on Potsdam $4\times$. All six paired-image bootstrap 95% confidence intervals lie strictly above zero, providing consistent image-level evidence that semantic conditioning benefits the evaluated fixed model pairs.

These intervals quantify variation over paired test images rather than uncertainty over training randomness. Because dataset and scale vary jointly between the two experiments, we interpret the results as evidence for the tested configurations rather than as establishing same-scale cross-dataset generality.

Table 15: **Refinement of the final seed-618 DirectSR candidate under $4\times$ SR.** All values are computed from saved uint8 RGB outputs using the unified full-image evaluator.

Dataset	Output	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓
Potsdam	DirectSR candidate	**34.19417**	0.867306	0.266076	0.170299	**52.4978**
	Candidate-aware	34.17898	**0.867413**	**0.262363**	**0.169310**	54.2052
	+DINO-FiLM	34.17411	0.867301	0.262929	0.169359	53.9924
Toronto	DirectSR candidate	29.10912	0.861453	0.196211	0.164305	**88.0307**
	Candidate-aware	**29.16620**	**0.862584**	**0.194077**	**0.163226**	88.0680
	+DINO-FiLM	29.11303	0.861607	0.194948	0.163779	88.1952

Table 16: **Paired-image bootstrap analysis for final-candidate refinement.** Values are mean differences relative to the final Stage-1 candidate; 95% bootstrap confidence intervals are shown below.

Dataset	Refiner	ΔPSNR	ΔLPIPS	ΔGrad-L1 ($\times 10^{-5}$)	ΔHF-L1 ($\times 10^{-5}$)	ΔEdge PSNR
Potsdam	Candidate-aware	-0.01520 $[-0.01652, -0.01391]$	-0.003713 $[-0.003835, -0.003593]$	$+0.803$ $[0.744, 0.861]$	$+1.93$ $[1.85, 2.02]$	$+0.01962$ $[0.01821, 0.02100]$
	+DINO-FiLM	-0.02006 $[-0.02103, -0.01913]$	-0.003147 $[-0.003248, -0.003047]$	$+1.15$ $[1.11, 1.19]$	$+2.23$ $[2.17, 2.30]$	$+0.00191$ $[0.00091, 0.00288]$
Toronto	Candidate-aware	$+0.05708$ $[0.05425, 0.06001]$	-0.002134 $[-0.002216, -0.002053]$	-6.25 $[-6.57, -5.94]$	-6.81 $[-7.20, -6.43]$	$+0.08380$ $[0.08064, 0.08707]$
	+DINO-FiLM	$+0.00390$ $[0.00283, 0.00504]$	-0.001263 $[-0.001309, -0.001219]$	-0.344 $[-0.458, -0.236]$	-0.175 $[-0.339, -0.018]$	$+0.01400$ $[0.01287, 0.01522]$

B.7 Final-Candidate DirectSR+ Analysis

To avoid conflating the earlier controlled ablation candidate with the final DirectSR output used in the main paper, we first evaluate Stage 2 refinement directly on the final seed-618 Stage-1 candidate.

This experiment constitutes our primary Stage 2 evidence because the refinement input is exactly the DirectSR candidate reported in the main $4\times$ reconstruction results. We compare the unrefined Stage-1 candidate, the candidate-aware RGB refiner, and the candidate-aware refiner augmented with DINO-FiLM. All metrics are computed from saved uint8 RGB outputs using the same full-image evaluator.

We further quantify paired image-level changes using $50{,}000$ bootstrap resamples of the fixed test images. Metric differences are defined as the refinement output minus the Stage-1 candidate. Positive differences indicate improvement for PSNR and Edge PSNR, whereas negative differences indicate improvement for LPIPS, Gradient-L1, and HF-L1.

FID is excluded from the paired bootstrap analysis because it is a dataset-level distributional metric and does not provide per-image paired observations under this protocol.

The final-candidate experiment reveals two distinct dataset behaviors.

On Potsdam, both refiners significantly reduce LPIPS and also improve DISTS, but PSNR decreases. Gradient-L1, HF-L1, and the LH/HL/HH directional high-frequency errors generally worsen. The refinement therefore represents a fidelity–perception trade-off rather than a uniform improvement in reconstruction fidelity.

On Toronto, candidate-aware refinement provides the clearest benefit: it improves PSNR by 0.05708 dB while also improving LPIPS, Gradient-L1, HF-L1, and Edge PSNR, with the corresponding paired-image bootstrap intervals separated from zero.

DINO-FiLM yields only a 0.00390 dB PSNR gain on Toronto and does not uniformly improve all directional high-frequency metrics, including a slight degradation in HH-L1. It should therefore not be interpreted as providing a comprehensive structural improvement.

Overall, the final-candidate analysis does not identify DINO-FiLM as a uniformly optimal $4\times$ Stage 2 configuration. DirectSR Stage 1 remains the default model, while the tested Stage 2 variants are treated as optional refinements.

Table 17: **Separate multi-branch refinement ablation under $4\times$ SR.** All values are RGB PSNR (dB). Within each dataset, all variants use the same fixed Stage-1 ablation candidate and test split. The table is intended only for controlled within-suite comparisons; its absolute values should not be compared with the final seed-618 candidate in Table 15.

Variant	Potsdam	Toronto	AID
Frozen Stage-1 candidate	34.10657	28.99993	**30.26966**
Candidate-aware	34.13837	29.03522	30.24287
+DINO-FiLM	34.14518	**29.04470**	30.24535
+Wavelet	34.13530	29.03078	30.24833
+DINO-FiLM+Wavelet	**34.14623**	29.03849	30.26051

B.7.1 Separate Multi-Branch Ablation Suite

In addition to the final-candidate analysis, we retain a separate multi-branch refinement suite to isolate candidate-aware, DINO-FiLM, and wavelet mechanisms under a common controlled input.

Importantly, this suite uses a separate fixed Stage-1 ablation candidate rather than the final seed-618 DirectSR candidate reported in Table 1. Its absolute candidate and refinement values should therefore not be compared directly with the final-candidate results. The experiment is intended only for controlled mechanism comparisons among variants within the suite.

Candidate-aware denotes the RGB refiner without DINO-FiLM or a wavelet-output head. *+DINO-FiLM* adds semantic modulation to the candidate-aware refiner. *+Wavelet* adds the learned wavelet-output branch, while *+DINO-FiLM+Wavelet* combines both extensions. These within-suite rankings are not interpreted as defining a universally preferred DirectSR+ configuration.

Within this separate ablation suite, all four refinement variants improve over the frozen candidate on Potsdam by 0.03180, 0.03861, 0.02873, and 0.03966 dB, respectively. On Toronto, the corresponding gains are 0.03529, 0.04477, 0.03085, and 0.03856 dB.

AID exhibits a different behavior: every refinement variant remains slightly below the frozen candidate, with differences ranging from -0.02679 to -0.00915 dB.

Isolating DINO-FiLM relative to the candidate-aware refiner gives gains of 0.00681, 0.00948, and 0.00248 dB on Potsdam, Toronto, and AID, respectively. Thus, under this particular fixed ablation candidate, semantic modulation provides a consistent but small improvement to the refiner itself.

Wavelet augmentation is less consistent. Used alone, it slightly decreases PSNR relative to the candidate-aware refiner on Potsdam and Toronto while improving it on AID. Combined with DINO-FiLM, it gives the highest PSNR within this suite on Potsdam but is not uniformly optimal across datasets.

Crucially, the final-candidate experiment in Table 15 shows that this within-suite ranking does not transfer uniformly to the final Stage-1 output. We therefore use this experiment only for controlled mechanism analysis, rather than to define a globally preferred DirectSR+ configuration. DirectSR Stage 1 remains the core default model, while candidate-aware, DINO-FiLM, and wavelet-based Stage 2 variants are treated as setting-dependent optional extensions.

B.8 Matched Full-Pipeline Comparisons

The main comparison in Sec. 4.4 isolates the Stage-1 recovery formulation by comparing matched trajectory-based Flow with endpoint-trained DirectSR. Here we report the corresponding full-pipeline comparisons, including trajectory-based Stage-1 models followed by their refinement stages. These results therefore complement rather than duplicate the Flow-step comparison in the main paper.

B.9 Complete DirectSR Metrics

For completeness, we report the full reconstruction, perceptual, and structural metrics for the six principal DirectSR/DirectSR+ evaluation settings. Each row corresponds to one complete unified-

Table 18: **Matched trajectory-based full-pipeline comparisons.**

Setting	Potsdam PSNR	Potsdam SSIM	Toronto PSNR	Toronto SSIM	AID PSNR	AID SSIM
Matched 4-step Flow + Refiner, 2×	41.65047	0.973808	36.86859	0.969637	36.30430	0.949070
Matched 4-step Flow, 4×	32.18770	0.840900	26.44330	0.792800	28.87730	0.777500
Matched 4-step Flow + Full Refiner, 4×	33.97958	0.865314	28.75687	0.856651	30.08254	0.815903

Table 19: **Complete reconstruction and perceptual metrics.**

Dataset	Output	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓	MUSIQ↑	CLIP-IQA+↑
Potsdam 2×	DirectSR+ seed618	41.92427	0.974871	0.052080	0.051607	7.3949	41.0944	0.437036
Toronto 2×	DirectSR+ seed618	37.31145	0.971140	0.040573	0.061315	15.7815	45.2936	0.557735
AID 2×	DirectSR seed618	36.41090	0.949958	0.087328	0.090776	27.8554	42.7024	0.563595
Potsdam 4×	DirectSR seed618	34.19417	0.867306	0.266076	0.170299	52.4978	34.6392	0.408730
Toronto 4×	DirectSR seed618	29.10912	0.861453	0.196211	0.164305	88.0307	40.2326	0.466330
AID 4×	DirectSR seed618	30.30181	0.821338	0.280806	0.194229	84.3711	37.8454	0.479960

Table 20: **Complete structural metrics.**

Dataset	Output	Grad-L1↓	HF-L1↓	LH-L1↓	HL-L1↓	HH-L1↓	Edge PSNR↑
Potsdam 2×	DirectSR+ seed618	0.003694	0.006233	0.007303	0.006736	0.004660	47.3826
Toronto 2×	DirectSR+ seed618	0.005848	0.009843	0.010409	0.011624	0.007494	42.7439
AID 2×	DirectSR seed618	0.007911	0.013206	0.015106	0.014961	0.009549	42.2732
Potsdam 4×	DirectSR seed618	0.007534	0.009604	0.011660	0.011233	0.005919	38.0182
Toronto 4×	DirectSR seed618	0.012367	0.015761	0.018151	0.019356	0.009777	32.9212
AID 4×	DirectSR seed618	0.013980	0.017820	0.021430	0.020954	0.011078	34.3253

evaluator run. The full seed-618 metrics are not combined with three-seed PSNR means into a single artificial result.