LEARNING TO TRUST GENERATIVE FEATURES FOR SPIKE IMAGE RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Spike cameras encode dynamic scenes as binary firing streams with ultra-high temporal resolution and high dynamic range. Recovering fine details from spike sequences remains challenging for reconstruction models based on direct regression. Generative restoration can provide complementary information, but its reliability varies across image regions. We propose LTGF, a framework that learns to trust generative features to enhance a base regression network. Conditioned on base spike features, a shared flow matching network starts from coarse intensity estimates based on spike firing rates (TFP) and inter-spike intervals (TFI) to generate complementary candidates. To select useful candidate information, multi-level uncertainty estimates in log space are combined with a learned content-adaptive tolerance that accounts for reconstruction benefit. The resulting weights control residual feature corrections while retaining the base representation. Experiments demonstrate the best performance among the compared methods on synthetic spike data and real spike data.

1 INTRODUCTION

Conventional frame cameras form images by integrating incident light over a predefined exposure interval. Although this sensing mechanism captures dense intensity information, it introduces an inherent trade-off between motion blur and signal-to-noise ratio. A long exposure improves photon efficiency but blurs rapidly moving objects, whereas a short exposure preserves motion at the cost of weak and noisy signals. Spike cameras provide an alternative paradigm for high-speed visual perception. Inspired by retinal photoreceptors, each pixel independently accumulates incoming light and emits a binary spike once the accumulated intensity reaches a preset threshold. Dynamic scenes are therefore recorded as continuous binary firing streams with ultra-high temporal resolution and high dynamic range, making spike cameras particularly suitable for capturing fast motion and challenging illumination conditions (Zhu et al., 2020; Zhao et al., 2021b; Zhu et al., 2021).

Recovering conventional intensity images from spike streams, however, remains a highly ill-posed inverse problem. Unlike frame-based cameras, spike cameras do not directly measure instantaneous intensity values; the latent image must be inferred from temporal spike patterns. Early non-learning methods use hand-crafted spike statistics. In particular, TFP estimates intensity from firing density within a temporal window, whereas TFI relies on local inter-spike intervals (Zhu et al., 2019; Zhao et al., 2020). Learning-based methods subsequently learn spike-to-image mappings with convolutional, recurrent, spatial-temporal architectures, and motion alignment to exploit spike sequences (Zhao et al., 2021b; Chen et al., 2023; Zhao et al., 2020; Zhang et al., 2023; Zhao et al., 2024).

These developments have substantially improved the recovery of global appearance and scene structure. Nevertheless, most existing methods remain centered on direct spike-to-image prediction. Stronger encoders and temporal modules improve the utilization of spikes, but the reconstruction is still inferred from a single regression model. When local spike evidence is sparse or ambiguous, such a model may favor conservative predictions that preserve dominant structures while attenuating uncertain high-frequency content, or it may produce unstable local artifacts around motion boundaries and fine textures.

Generative restoration models offer a different source of reconstruction information. Diffusion and flow-based models learn how degraded observations can be transformed toward clean image distributions, and their internal features encode structural correction, degradation removal, and detail

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.

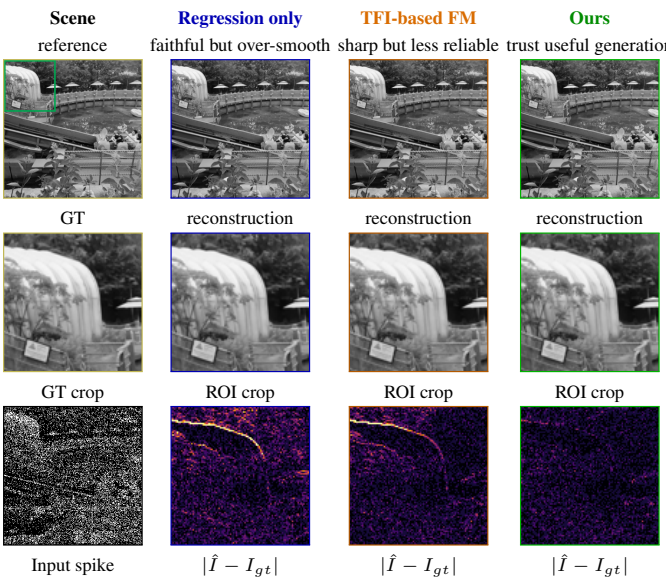

Figure 1: The left column shows GT, its crop, and input spikes. TFI-based FM supplies sharp but less reliable detail; LTGF selectively enhances the base regression network to reduce errors. Error maps show pixelwise absolute differences from GT.

formation (Rombach et al., 2022; Lipman et al., 2022; Liu et al., 2022; Ye et al., 2024). For spike reconstruction, these generative features can provide candidate information beyond that recovered by direct regression. Our goal is to use generative restoration knowledge as auxiliary information to refine the spike features of a regression model.

Once generative modeling is introduced as an auxiliary source, a practical question is what spike-specific inputs should drive it. Fortunately, two coarse intensity observations can be obtained directly from the temporal statistics of the input stream. TFP aggregates spikes over a temporal window and therefore provides a smooth observation that may suffer from motion blur, whereas TFI is computed from local inter-spike intervals and retains richer textures but is more sensitive to noise. These observations offer different views of the same spike sequence and serve as convenient starting states for a shared spike-conditioned flow model, from which useful generative restoration knowledge can be learned. However, the resulting generative features are not uniformly reliable: some may recover missing structures, while others may carry observation noise or unsupported content. The key is therefore to determine when and where they should assist regression-based reconstruction, rather than fusing them indiscriminately.

Based on this insight, we propose LTGF, an uncertainty-guided framework that learns when and where generative features should be trusted. A base regression network extracts stable hierarchical spike features, while a shared spike-conditioned flow model learns source-specific generative restoration knowledge from the TFP and TFI observations. An uncertainty-guided fusion module then selectively transfers reliable generative corrections into the base representation while suppressing unreliable information. In this way, generative features enhance rather than replace the base reconstruction. The entire framework is optimized jointly from scratch.

Fig. 1 provides a diagnostic comparison among the base regression network, a TFI-based single-source flow variant, and the full LTGF model. The base network produces stable reconstructions but leaves noticeable residual errors around fine boundaries and textured regions, while the single-source flow model recovers additional details but may also introduce unreliable corrections. By treating generative knowledge as auxiliary information and selectively transferring reliable features, LTGF further reduces localized reconstruction errors while preserving the stable content recovered from the spikes.

The main contributions are summarized as follows:

- We propose LTGF, a framework that learns when and where to use generative features to enhance a base regression network for spike image reconstruction.
- We exploit coarse observations that are readily derived from spike temporal statistics as source states for a shared spike-conditioned flow model, enabling task-specific generative knowledge to assist regression-based reconstruction.
- We introduce an uncertainty-guided trust mechanism that assesses source reliability and selectively transfers source-specific generative corrections into the base features.
- Experiments on synthetic and real spike data demonstrate improved reconstruction fidelity and perceptual quality over representative spike reconstruction methods.

2 Related Work

Spike camera image reconstruction. Early methods recover intensity through firing statistics or motion-aware temporal estimation (Zhu et al., 2020; Zhao et al., 2020). Learning-based approaches develop stronger spike representations through convolutional reconstruction, super-resolution, self-supervision, recurrence, and temporal-order modeling (Zhao et al., 2021b;a; Chen et al., 2022; Zhu et al., 2023; Zhang et al., 2023). Recent advances suppress spike fluctuations with temporal alignment (Zhao et al., 2024), jointly refine temporal and spatial information (Fan et al., 2024), model continuous-time dynamics (Yang et al., 2024), or exploit state-space representations (Yin et al., 2025); other work revisits the reconstruction paradigm (Chen et al., 2025). These methods primarily improve spike feature extraction or decoding. LTGF instead complements a base representation with generative features and estimates their reliability before fusion.

Generative restoration and flow matching. Diffusion models demonstrate strong image-generation capability (Rombach et al., 2022), while flow matching and rectified flow learn continuous vector fields for generation (Lipman et al., 2022; Liu et al., 2022). Generative priors also support texture recovery in restoration (Ye et al., 2024) and spike reconstruction under low illumination (Liu et al., 2025). LTGF trains FM jointly with the spike network from scratch, using TFP and TFI starting images to provide complementary restoration features. Uncertainty-guided residual fusion selects their contributions to the base features rather than directly adopting a generated image as the final reconstruction.

3 Method

3.1 Overview

Appendix A describes the spike camera imaging principle. Given a binary spike sequence $S = [S_1, \ldots, S_T] \in \{0,1\}^{T\times H\times W}$ with odd T, we reconstruct the intensity image $\hat{I}$ at its center frame $j_c = (T+1)/2$, supervised by I_{gt}. LTGF enhances spike features with generative features whose contribution is controlled by their estimated reliability. As shown in Fig. 2, the base network first extracts global and local temporal correlations from S, producing a feature pyramid $E = \{E^l\}_{l=0}^{L-1}$ and a base image $\hat{I}_b$. A flow matching (FM) network then uses TFP and TFI images as two different starting points to generate candidate images and features, conditioned on E. Uncertainty heads estimate the local prediction error scale at all four FM feature levels. These estimates guide the fusion of the candidate features into E, and a reconstruction decoder converts the enhanced pyramid $\bar{E}$ into $\hat{I}$. We use $L = 4$ feature levels, indexed by $l = 0, \ldots, L-1$ from finest to coarsest, with $E^l \in \mathbb{R}^{D\times H_l\times W_l}$ and $(H_l, W_l) = (H, W)/2^l$. Brackets $[\cdot]$ denote channel concatenation. Network configurations are provided in Appendix B.

3.2 Base Spike Feature Extraction

The base network provides spike features and an image reference for candidate corrections. The full-window encoder f_{fw} processes the entire sequence S to capture global temporal correlations. For local temporal correlations, we divide S into N overlapping blocks B_i of R frames and encode each with f_{blk}. Both encoders use a convolution and a residual block to produce D-channel features at the input resolution:

$$F_{fw} = f_{fw}(S), \qquad F_i = f_{blk}(B_i), \quad i = 1, \ldots, N. \tag{1}$$

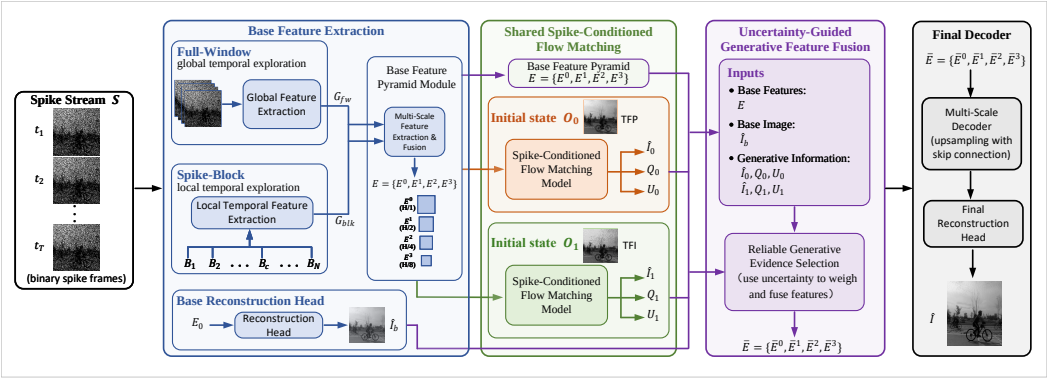

Figure 2: LTGF overview. Base spike features condition FM candidates from TFP and TFI; multi-level uncertainty guides their residual fusion for final reconstruction.

The middle block B_c, $c = (N+1)/2$, is centered at j_c. To aggregate local information around this target frame, local cross-attention f_{att} uses F_c as query and neighboring F_i as context. A residual fusion block f_{bf} combines the attended features:

$$\widetilde{F}_i = f_{att}(F_c, F_i), \quad i \neq c, \qquad F_{blk} = f_{bf}\Big([F_c, \{\widetilde{F}_i\}_{i\neq c}]\Big). \tag{2}$$

Residual refinement f_{ref} gives $G_a = f_{ref}(F_a)$ for $a \in \{fw, blk\}$. The refined full-window and local features form spatial pyramids, fused at corresponding resolutions:

$$\begin{aligned} &E_a^0 = G_a, \qquad E_a^l = f_{pyr}^l(E_a^{l-1}), \quad l = 1, \dots, L-1, \\ &E^l = f_{fuse}([E_{fw}^l, E_{blk}^l]), \quad l = 0, \dots, L-1. \end{aligned} \tag{3}$$

Here f_{pyr}^l combines residual downsampling and refinement blocks, and f_{fuse} is a residual fusion block. Thus, E incorporates both global and local temporal information. A convolutional image head f_{img} predicts the base image $\hat{I}_b$ from E^0 as a reference for base feature enhancement, with auxiliary predictions from G_{fw} and G_{blk}:

$$\hat{I}_{fw} = f_{img}(G_{fw}), \qquad \hat{I}_{blk} = f_{img}(G_{blk}), \qquad \hat{I}_b = f_{img}(E^0). \tag{4}$$

3.3 Spike-Conditioned Flow Matching

To complement the base features, FM generates restoration candidates from different starting points. TFP averages spikes over a window and can smooth temporal changes, whereas TFI uses the interval around the target time and is more sensitive to spike-timing fluctuations. These different temporal statistics give the fusion module multiple candidates while conditioning both branches on the same spike features E.

Starting images and restoration field. At pixel p, let j_p^- and j_p^+ be the spike indices immediately before or at j_c and after j_c. The TFP ($k = 0$) and TFI ($k = 1$) starting images are

$$O_0(p) = \mathcal{N}\left(\frac{1}{\eta T}\sum_{j=1}^{T} S_j(p)\right), \qquad O_1(p) = \mathcal{N}\left(\frac{1}{\eta(j_p^+ - j_p^-)}\right), \tag{5}$$

where η is the photoelectric conversion rate and $\mathcal{N}$ maps the estimates to the target intensity scale. During training, a sampled flow time $t \in [0, 1)$ defines a straight path and its target velocity:

$$x_{k,t} = (1-t)O_k + tI_{gt}, \qquad v_k^* = \frac{dx_{k,t}}{dt} = I_{gt} - O_k. \tag{6}$$

Flow time t denotes restoration progress, distinct from spike index j. Conditioned on E, the FM network predicts the required intensity change and a feature pyramid:

$$(\hat{v}_{k,t}, Q_{k,t}) = f_{fm}(x_{k,t}, t, s_k; E), \qquad Q_{k,t} = \{Q_{k,t}^l\}_{l=0}^{L-1}, \tag{7}$$

where $s_k = k$ identifies the source and $Q^l_{k,t}$ has the same dimensions as E^l.

The function f_{fm} is a four-level residual U-Net shared by the TFP and TFI branches. Its encoder concatenates $x_{k,t}$ with a spatially broadcast s_k and uses a sinusoidal time embedding to scale and shift residual features. Each level fuses its feature with E^l through a residual block, retaining spike conditioning; residual downsampling and refinement connect successive levels. The decoder uses residual upsampling, additive encoder skips, and refinement blocks. An output convolution predicts $\hat{v}_{k,t}$; the three decoded features and bottleneck feature form $Q_{k,t}$.

Single-step candidates. FM predicts velocity, so we use one Euler step to obtain a candidate image:

$$\hat{I}_{k,t} = x_{k,t} + (1-t)\hat{v}_{k,t}. \tag{8}$$

To enhance the base features, we generate each FM candidate with a single evaluation at $t = 0$ from O_k. This gives $\hat{I}_k = O_k + \hat{v}_{k,0}$ and features $Q^l_k = Q^l_{k,0}$. Their local reliability is estimated next to guide their fusion with the base features.

3.4 Multi-Level Uncertainty Estimation

Candidate reliability varies across regions and levels. We therefore apply a convolutional uncertainty head f^l_{unc} to each level of $Q_{k,t}$, including the bottleneck. Its linear output directly predicts a *log velocity-error variance* map:

$$V^l_{k,t} = f^l_{unc}(Q^l_{k,t}) \in \mathbb{R}^{1\times H_l \times W_l}. \tag{9}$$

Thus, $\exp(V^l_{k,t})$ represents a positive error-variance estimate. The target for $V^l_{k,t}$ is the local squared velocity residual in log space:

$$R_{k,t} = (\mathrm{sg}(\hat{v}_{k,t}) - v^*_k)^2, \qquad \widetilde{V}^l_{k,t} = \log(\mathrm{Pool}_l(R_{k,t}) + \epsilon)\,. \tag{10}$$

Here sg stops gradients through the target, Pool_l averages to resolution $H_l \times W_l$, and $\epsilon > 0$ keeps the logarithm finite. Pooling before the logarithm yields a level-specific estimate of local mean squared error.

Fusion requires uncertainty on the image-error scale. Subtracting $I_{gt} = x_{k,t} + (1-t)v^*_k$ from Eq. (8) gives

$$\hat{I}_{k,t} - I_{gt} = (1-t)(\hat{v}_{k,t} - v^*_k). \tag{11}$$

Squared image error therefore scales by $(1-t)^2$, which becomes an additive conversion in log space:

$$U^l_{k,t} = \log\big((1-t)^2 \exp(V^l_{k,t})\big) = V^l_{k,t} + 2\log(1-t), \qquad t < 1. \tag{12}$$

The log image-error variance $U^l_{k,t}$ needs no additional head. At the fusion time $t = 0$, $U^l_k = U^l_{k,0} = V^l_{k,0}$: candidate features and uncertainty come from the same FM evaluation.

3.5 Uncertainty-Guided Residual Fusion

With these uncertainty estimates, fusion separates what a candidate can add from how much to trust it: a correction network predicts an update to the base features, and an uncertainty-guided weight controls its contribution (Fig. 3).

Candidate corrections. To identify the details proposed by a branch, we resize its image difference from the base and remove the local mean. A convolutional embedding f_{res} and adapter f^l_{prop} incorporate this detail signal into the FM feature:

$$J^l_k = f_{res}\Big(\mathrm{HP}\Big(\mathrm{Resize}_l(\hat{I}_k - \hat{I}_b)\Big)\Big), \qquad P^l_k = Q^l_k + f^l_{prop}(J^l_k). \tag{13}$$

Here Resize_l matches the spatial resolution at level l, and HP subtracts the local mean. From the base feature E^l and proposal P^l_k, a convolutional network f^l_{corr} learns a residual correction Δ^l_k:

$$\Delta^l_k = f^l_{corr}\big([\mathrm{LN}(E^l), \mathrm{LN}(P^l_k)]\big)\,, \tag{14}$$

where LN normalizes feature channels at each pixel and Δ^l_k is a D-channel update to E^l.

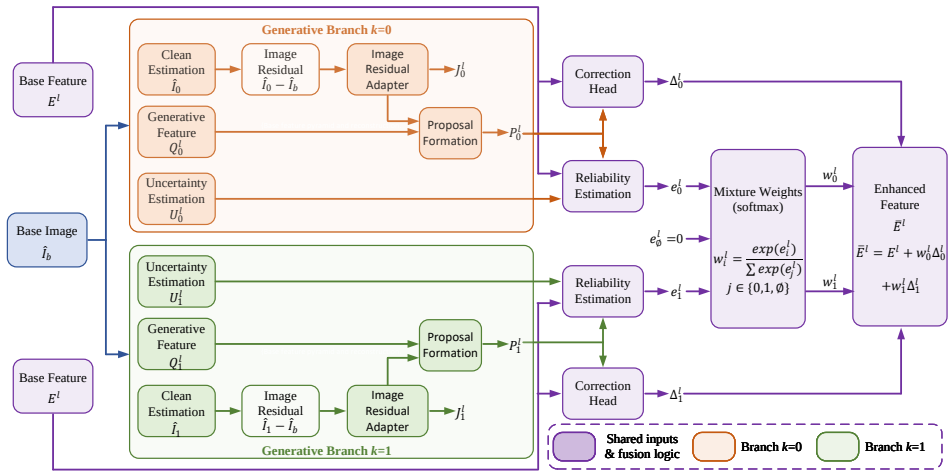

Figure 3: Residual fusion at level l. Log uncertainty U_k^l and content-adaptive tolerance τ_k^l weight candidate corrections; a base expert controls their total contribution.

Confidence in log space. Image uncertainty alone does not determine how useful candidate features are relative to the base. A content-adaptive tolerance τ_k^l learns, through reconstruction supervision, to compensate for the mismatch between uncertainty and actual fusion benefit. f_{tol}^l predicts this adjustment from the base and proposal features:

$$\tau_k^l = f_{tol}^l\big([\mathrm{LN}(E^l), \mathrm{LN}(P_k^l)]\big)\,,$$
$$e_k^l = \log\left(\frac{\exp(\tau_k^l)}{\sqrt{\exp(U_k^l)}}\right) = \tau_k^l - \frac{1}{2}U_k^l. \tag{15}$$

The spatial fusion logit e_k^l is an unnormalized confidence score. Since U_k^l is a log variance, $\sqrt{\exp(U_k^l)} = \exp(U_k^l/2)$ is the predicted image-error standard-deviation scale. Its inverse supplies uncertainty-based confidence, adjusted by the positive factor $\exp(\tau_k^l)$. Taking the logarithm explains the subtraction and factor $1/2$. Thus, higher uncertainty lowers confidence for fixed tolerance, while learned tolerance accounts for candidate usefulness.

Preserving the base. A two-branch softmax would always allocate a total weight of one to FM corrections. We add the unchanged base as a third expert with zero logit:

$$[w_\varnothing^l, w_0^l, w_1^l] = \mathrm{softmax}([0, e_0^l, e_1^l]). \tag{16}$$

When both FM logits decrease, the base weight $w_\varnothing^l$ increases, reducing their total contribution. The enhanced feature is

$$\bar{E}^l = w_\varnothing^l E^l + \sum_{k=0}^{1} w_k^l(E^l + \Delta_k^l) = E^l + \sum_{k=0}^{1} w_k^l \Delta_k^l. \tag{17}$$

The equality follows from weights summing to one. These spatial weights are broadcast over channels, allowing selective enhancement while retaining the base reference. A residual upsampling decoder f_{dec} with skips from all enhanced levels produces $\hat{I} = f_{dec}(\bar{E})$.

3.6 Training Objective

We jointly train all modules from scratch. Velocity and uncertainty are supervised at both $r \in \{t, 0\}$: the sampled flow time t and the starting time 0. Final and base images use mean L_1 losses:

$$\mathcal{L}_{rec} = \frac{1}{M}\|\hat{I} - I_{gt}\|_1, \qquad \mathcal{L}_{base} = \frac{1}{3M}\sum_{a\in\{fw,blk,b\}} \|\hat{I}_a - I_{gt}\|_1, \tag{18}$$

Table 1: Synthetic reconstruction (best in bold; second best underlined).

Method	PSNR↑	SSIM↑	LPIPS↓
TFI	24.0299	0.6993	0.3712
TFP	29.2620	0.8029	0.3370
TFSTP	23.3161	0.6633	0.4032
Spk2Img	41.7807	0.9852	0.0423
SpikeFormer	41.4812	0.9849	0.0388
SSML	35.8808	0.9600	0.1163
WGSE	41.4236	0.9850	0.0452
SpikeODE	<u>43.1874</u>	<u>0.9888</u>	<u>0.0321</u>
BSF	40.1795	0.9795	0.0575
S2I-Mamba	38.0498	0.9710	0.0768
Ours	**43.4765**	**0.9896**	**0.0299**

Table 2: Average no-reference quality on six real scenes. Formatting follows Table 1.

Method	HyperIQA↑	MUSIQ↑	MANIQA↑
TFI	0.4644	66.0076	0.5232
TFP	0.3415	61.2541	0.5331
TFSTP	0.4627	65.9784	0.4963
Spk2Img	0.5283	67.7449	0.6189
SpikeFormer	0.5085	67.2269	0.6033
SSML	0.4144	65.6534	0.6006
WGSE	0.5129	67.4852	0.6127
SpikeODE	<u>0.5437</u>	<u>68.0292</u>	<u>0.6290</u>
BSF	0.4917	66.8873	0.6031
S2I-Mamba	0.4618	65.6807	0.5883
Ours	**0.5510**	**68.1292**	**0.6350**

Here, M counts pixels across the batch, and tensors are vectorized. Velocity and uncertainty losses are:

$$\begin{aligned}\mathcal{L}_{fm} &= \frac{1}{2M}\sum_{k=0}^{1}\sum_{r\in\{t,0\}}\|\hat{v}_{k,r}-v_k^*\|_2^2,\\ \mathcal{L}_{unc} &= \frac{1}{2L}\sum_{k=0}^{1}\sum_{l=0}^{L-1}\sum_{r\in\{t\}}\frac{1}{M_l}\|V_{k,r}^l-\widetilde{V}_{k,r}^l\|_1.\end{aligned} \tag{19}$$

Here, M_l counts pixels across the batch at level l, so uncertainty uses mean L_1 loss on log variance. Then The total loss is:

$$\mathcal{L}=\lambda_{rec}\mathcal{L}_{rec}+\lambda_{base}\mathcal{L}_{base}+\lambda_{fm}\mathcal{L}_{fm}+\lambda_{unc}\mathcal{L}_{unc}. \tag{20}$$

4 Experiments

4.1 Experimental Setup

Paired ground truth image is unavailable for the real spike data, so we train on synthetic data. Following Zhao et al. (2021b); Yang et al. (2024), we use a REDS-based dataset (Nah et al., 2019) pairing simulated spike sequences with ground truth images, split into 800 training, 80 validation, and 80 test pairs at 256×256. We also test on real spike data from six scenes: *Car*, *Rotation-White*, *Rotation-Black*, *Railway*, *Train*, and *Bridge*. We report PSNR, SSIM, and LPIPS (Zhang et al., 2018) on synthetic data, and no-reference HyperIQA (Su et al., 2020), MUSIQ (Ke et al., 2021), and MANIQA (Yang et al., 2022) on real data. Higher scores are better except for LPIPS.

Implementation details and additional scene-wise results are provided in Appendices B.4 and D, respectively.

4.2 Comparison with Existing Methods

We compare with TFI, TFP (Zhu et al., 2019), TFSTP (Zheng et al., 2021), Spk2Img (Zhao et al., 2021b), SpikeFormer (She & Qing, 2022), SSML (Chen et al., 2022), WGSE (Zhang et al., 2023), SpikeODE (Yang et al., 2024), BSF (Zhao et al., 2024), and S2I-Mamba (Yin et al., 2025).

Synthetic data. LTGF achieves the highest PSNR and SSIM and the lowest LPIPS in Table 1, gaining 0.2891 dB over SpikeODE. Figure 4 pairs reconstruction crops with pixelwise absolute-error maps on a common 0–10 gray-level display range. Smaller errors around thin boundaries, repeated structures, and textures complement the quantitative gains in fidelity and perceptual similarity.

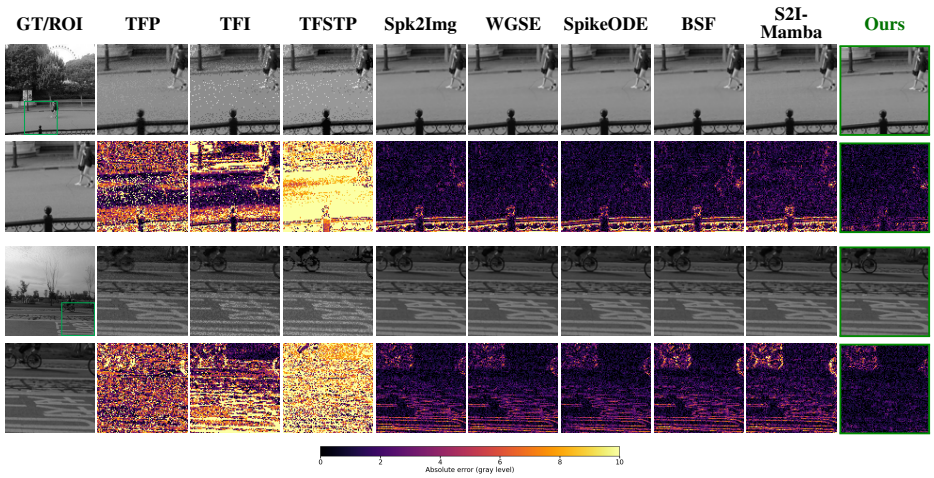

Figure 4: Synthetic reconstruction crops (upper rows) and absolute-error maps (lower rows). Darker colors indicate smaller errors; values above 10 gray levels are clipped only for display.

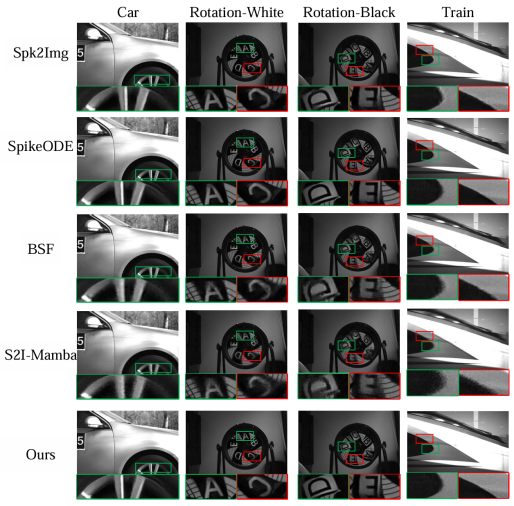

Figure 5: Qualitative comparison on the real scenes *Car*, *Rotation-White*, *Rotation-Black*, *Train*.

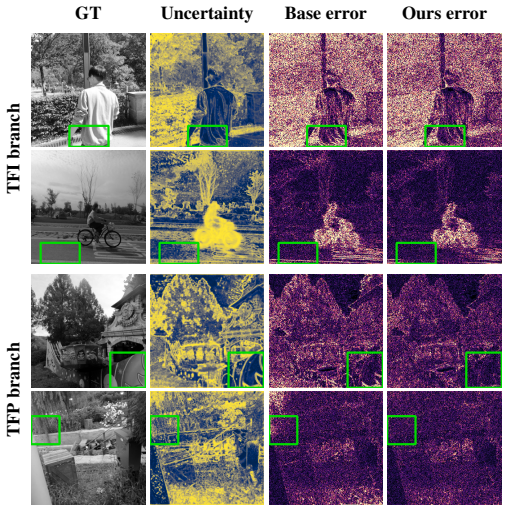

Figure 6: Uncertainty and reconstruction errors for the TFI and TFP branches. Green boxes mark corresponding regions.

Real data. LTGF obtains the best averages on all three no-reference metrics (Table 2), with small margins over SpikeODE. Figure 5 shows improved preservation of fine boundaries and local structures in the car, rotation, and train scenes, supporting the quantitative trend.

4.3 Ablation Study

Generative sources. Table 3 shows that either FM source alone underperforms the base, yet fusing either source with the base improves all three metrics. This supports using FM as an auxiliary source of restoration features. Combining both sources performs best, gaining 2.5791 dB over the base and 0.5447 dB over the stronger single-source variant. Their complementary features are therefore useful when selectively fused with the spike representation.

Fusion strategy. Table 4 compares the base, feature-attention fusion, concatenation, and full LTGF. All source-assisted variants use the same base extractor, two FM sources, proposals, decoder, and training protocol. Feature attention predicts logits directly from base and proposal features, retaining the base expert and residual corrections but omitting the explicit uncertainty term. Concatenation

Table 3: Generative-source ablation. Source 0: TFP; source 1: TFI.

Variant	PSNR↑	SSIM↑	LPIPS↓
Base pathway	40.8974	0.9825	0.0609
FM source 0 only	34.8722	0.9460	0.1259
FM source 1 only	35.1086	0.9510	0.0990
Base + source 0	42.8880	0.9878	0.0407
Base + source 1	42.9318	0.9878	0.0407
Ours (full)	**43.4765**	**0.9896**	**0.0299**

Table 4: Fusion-strategy ablation. Formatting follows Table 1.

Variant	PSNR↑	SSIM↑	LPIPS↓
Base model	40.8974	0.9825	0.0609
Feature-attention fusion	43.1851	0.9878	0.0384
Concatenation fusion	43.0847	0.9877	0.0387
Ours (full)	**43.4765**	**0.9896**	**0.0299**

Table 5: Reconstruction quality and model complexity. PSNR is reported on the synthesized benchmark; complexity is measured for one 45-frame input at 256×400, with synchronized timing excluding model loading.

Method	PSNR↑	Params. (M)↓	FLOPs (T)↓	Time (s)↓
Spk2Img	41.7807	3.8765	2.3055	0.5970
SpikeFormer	41.4812	7.5699	0.0651	0.0648
SSML	35.8808	3.6250	0.6310	0.1904
WGSE	41.4236	3.8148	0.4172	0.1118
SpikeODE	43.1874	6.0731	2.3209	0.2981
BSF	40.1795	2.4769	0.7258	0.1743
S2I-Mamba	38.0498	**0.3420**	**0.0106**	**0.0158**
Ours	**43.4765**	3.2563	0.3784	0.1093

combines the three features through a channel-reducing convolution without adaptive weights; Appendix C provides the definitions. Both alternatives improve over the base, while LTGF performs best on all metrics. Its advantage over feature attention supports explicitly incorporating uncertainty; the comparison with concatenation supports selective fusion more broadly.

Spatial uncertainty analysis. Figure 6 shows GT, branch uncertainty, base error, and full-model error for two examples per branch. Errors are pixelwise absolute differences from GT; uncertainty is displayed as percentile rank. Brighter colors mean higher uncertainty or larger error, while darker colors mean lower uncertainty or smaller error. In the green boxes, low-uncertainty regions coincide with reduced reconstruction errors: on the garment and road for TFI, and the curved surface and tree-trunk/wall region for TFP. Residual errors remain, indicating spatially selective improvement. These correspondences support uncertainty as a reliability cue alongside the fusion ablation.

4.4 Complexity Analysis

LTGF attains the highest PSNR in Table 5 with 3.2563M parameters, 0.3784T FLOPs, and 0.1093 s inference time. It gains 0.2891 dB over SpikeODE while reducing parameters, FLOPs, and runtime by 46.4%, 83.7%, and 63.3%, respectively. S2I-Mamba is more efficient but has 5.4267 dB lower PSNR, illustrating the quality–complexity trade-off.

5 Conclusion

We presented LTGF, a framework that uses generative features to enhance a base regression network for spike image reconstruction. Complementary TFP and TFI starting points provide diverse restoration candidates, while multi-level uncertainty and learned tolerance control their residual contributions. Experiments show improved fidelity and perceptual similarity on synthetic data and higher average no-reference quality scores on real data. Ablations support both source complementarity and uncertainty-guided selection, demonstrating the benefit of selectively incorporating generative information while preserving the base representation.

REFERENCES

Kang Chen, Yajing Zheng, Tiejun Huang, and Zhaofei Yu. Rethinking high-speed image reconstruction framework with spike camera. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 2097–2104, 2025.

Shiyan Chen, Chaoteng Duan, Zhaofei Yu, Ruiqin Xiong, and Tiejun Huang. Self-supervised mutual learning for dynamic scene reconstruction of spiking camera. In *IJCAI*, pp. 2859–2866, 2022.

Shiyan Chen, Zhaofei Yu, and Tiejun Huang. Self-supervised joint dynamic scene reconstruction and optical flow estimation for spiking camera. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 350–358, 2023.

Bin Fan, Jiaoyang Yin, Yuchao Dai, Chao Xu, Tiejun Huang, and Boxin Shi. Spatio-temporal interactive learning for efficient image reconstruction of spiking cameras. *Advances in Neural Information Processing Systems*, 37:21401–21427, 2024.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5128–5137. IEEE, 2021.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Ruonan Liu, Lin Zhu, Xijie Xiang, Lizhi Wang, and Hua Huang. Noise-modeled diffusion models for low-light spike image restoration. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4080–4089. IEEE, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1996–2005. IEEE, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Chen She and Laiyun Qing. Spikeformer: Image reconstruction from the sequence of spike camera based on transformer. In *Proceedings of the 2022 5th International Conference on Image and Graphics Processing*, pp. 72–78, 2022.

Shaolin Su, Qingsen Yan, Yu Zhu, Cheng Zhang, Xin Ge, Jinqiu Sun, and Yanning Zhang. Blindly assess image quality in the wild guided by a self-adaptive hyper network. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3664–3673. IEEE, 2020.

Chen Yang, Guorong Li, Shuhui Wang, Li Su, Laiyun Qing, and Qingming Huang. Spikeode: Image reconstruction for spike camera with neural ordinary differential equation. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(11):11142–11155, 2024.

Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao, Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1190–1199. IEEE, 2022.

Tian Ye, Sixiang Chen, Wenhao Chai, Zhaohu Xing, Jing Qin, Ge Lin, and Lei Zhu. Learning diffusion texture priors for image restoration. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2524–2534. IEEE, 2024.

Jiaoyang Yin, Bin Fan, Chao Xu, Tiejun Huang, and Boxin Shi. Spk2imgmamba: spiking camera image reconstruction with multi-scale state space models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 9342–9352. IEEE, 2025.

Jiyuan Zhang, Shanshan Jia, Zhaofei Yu, and Tiejun Huang. Learning temporal-ordered representation for spike streams based on discrete wavelet transforms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 137–147, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Jing Zhao, Ruiqin Xiong, and Tiejun Huang. High-speed motion scene reconstruction for spike camera via motion aligned filtering. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5. IEEE, 2020.

Jing Zhao, Jiyu Xie, Ruiqin Xiong, Jian Zhang, Zhaofei Yu, and Tiejun Huang. Super resolve dynamic scene from continuous spike streams. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2513–2522. IEEE, 2021a.

Jing Zhao, Ruiqin Xiong, Hangfan Liu, Jian Zhang, and Tiejun Huang. Spk2imgnet: Learning to reconstruct dynamic scene from continuous spike stream. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11991–12000. IEEE, 2021b.

Rui Zhao, Ruiqin Xiong, Jing Zhao, Jian Zhang, Xiaopeng Fan, Zhaofei Yu, and Tiejun Huang. Boosting spike camera image reconstruction from a perspective of dealing with spike fluctuations. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24955–24965. IEEE, 2024.

Yajing Zheng, Lingxiao Zheng, Zhaofei Yu, Boxin Shi, Yonghong Tian, and Tiejun Huang. High-speed image reconstruction through short-term plasticity for spiking cameras. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6354–6363. IEEE, 2021.

Lin Zhu, Siwei Dong, Tiejun Huang, and Yonghong Tian. A retina-inspired sampling method for visual texture reconstruction. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1432–1437. IEEE, 2019.

Lin Zhu, Siwei Dong, Jianing Li, Tiejun Huang, and Yonghong Tian. Retina-like visual image reconstruction via spiking neural model. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1435–1443. IEEE, 2020.

Lin Zhu, Jianing Li, Xiao Wang, Tiejun Huang, and Yonghong Tian. Neuspike-net: High speed video reconstruction via bio-inspired neuromorphic cameras. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2380–2389. IEEE, 2021.

Lin Zhu, Yunlong Zheng, Mengyue Geng, Lizhi Wang, and Hua Huang. Recurrent spike-based image restoration under general illumination. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 8251–8260, 2023.

Appendix

This appendix introduces the spike camera imaging principle and provides implementation details and additional evidence for LTGF. Section A explains how light is encoded into binary spikes and relates this process to the TFP and TFI starting images. Section B describes the network architecture, input construction, and training settings. Section C defines the feature-attention and concatenation variants used in the fusion ablation. Section D presents extended synthetic-data visualizations, scene-wise no-reference scores, and complete qualitative comparisons on real spike data. Section E examines the base, TFP, and TFI fusion weight distributions across feature levels. Finally, Section F evaluates the relationship between predicted uncertainty and FM candidate error over the synthetic test set, including error ordering and agreement in magnitude.

A Spike Camera Imaging Principle

A spike camera consists of an $H \times W$ pixel array. Each pixel p independently converts incident light of intensity $I(p, \xi)$ into electric charge and accumulates it over time. Here ξ denotes acquisition time, distinct from the flow time t in our method. When the accumulated charge reaches a threshold θ, the pixel emits a spike and resets its charge to zero. Under this ideal integrate-and-fire model with zero initial charge, the residual charge is

$$A(p, \xi) = \left(\eta \int_0^{\xi} I(p, u)\, du \right) \bmod \theta, \tag{21}$$

where η is the photoelectric conversion rate, using the same notation as Eq. (5), and the modulo operation accounts for repeated firing and reset.

The camera periodically reads the binary firing status of each pixel: $S_j(p) = 1$ if a spike is recorded in sampling interval j, and 0 otherwise. This produces $S = [S_1, \ldots, S_T] \in \{0, 1\}^{T \times H \times W}$, where T is the number of sampled spike frames. The intensity estimates in Eq. (5) use normalized charge and time units, with unit firing threshold and sampling interval. In these units, TFP estimates intensity from the mean firing rate divided by η, whereas TFI uses the reciprocal of the inter-spike interval divided by η. These two statistics provide the complementary starting images for the shared FM network.

B Implementation Details

This section details the base and FM networks, uncertainty heads, fusion modules, and final decoder, followed by input construction and training settings.

B.1 Base Network

All four feature levels use $D = 64$ channels. A residual block contains two 3×3 convolutions, LeakyReLU activations, and a shortcut, with a 1×1 projection when channel dimensions differ. The full-window and block encoders, f_{fw} and f_{blk}, each comprise a 3×3 convolution, LeakyReLU, and a residual block. Input spike frames are stacked along the channel dimension.

The local cross-attention module f_{att} uses a single head over a 3×3 context neighborhood. Learned queries and keys are formed from normalized features, and scaled dot-product attention aggregates context values before an output projection. The residual fusion block f_{bf} reduces the concatenated ND channels to D. The refinement network f_{ref} contains two residual blocks. Each pyramid stage f_{pyr}^{l} uses a residual downsampling block with stride 2 followed by a refinement residual block, and f_{fuse} maps $2D$ channels to D. The image head f_{img} contains three residual blocks and a 3×3 output convolution. It is applied to G_{fw}, G_{blk}, and E^0, with all three outputs supervised equally.

B.2 FM Network and Uncertainty Heads

The FM input concatenates the image state with the spatially broadcast source indicator, followed by a 3×3 convolution and a time-conditioned residual block. Each encoder level concatenates

its feature with E^l and applies a residual fusion block. Three residual downsampling stages, each followed by refinement, connect the four levels. The decoder uses three residual upsampling blocks with transposed convolutions, additive encoder skips, and residual refinement. A 3×3 convolution maps the finest feature to velocity. The pyramid $Q_{k,t}$ contains three decoded features ($l = 0, 1, 2$) and the bottleneck feature ($l = 3$).

A 32-dimensional sinusoidal encoding of t is mapped to 64 channels by two fully connected layers, each followed by SiLU. Each encoder residual block projects this embedding to channel-wise scale and shift vectors $a(t)$ and $b(t)$. After its first convolution and before activation, a feature h is modulated as $h \odot (1 + a(t)) + b(t)$ with spatial broadcasting. This modulation applies to encoder input, fusion, downsampling, and refinement blocks; the decoder receives no additional time modulation.

At each level of $Q_{k,t}$, the uncertainty head f^l_{unc} contains a 1×1 convolution, a 3×3 convolution, and a 1×1 output convolution. LeakyReLU follows the first two layers, and the output is linear. The target in Eq. (10) uses adaptive average pooling and $\epsilon = 10^{-6}$. At each time, we average absolute log-variance errors within each level, then over levels and sources, and sum the two time losses. Detached targets block gradients through the velocity prediction while retaining gradients to uncertainty heads and FM features.

B.3 Fusion Modules and Final Decoder

The resize operator in Eq. (13) uses average pooling, and HP subtracts a 3×3 local mean. The residual embedding f_{res} is a 3×3 convolution with LeakyReLU. At each level, a proposal adapter f^l_{prop} contains a standard 3×3 convolution, LeakyReLU, and a 1×1 convolution. Channel normalization LN is applied independently at each pixel without learned affine parameters.

Each level-specific correction network f^l_{corr} uses 1×1, 3×3, and 1×1 convolutions, with LeakyReLU after the first two, mapping $2D$ input channels to a D-channel update. The tolerance head f^l_{tol} uses a 1×1 convolution from $2D$ to 16 channels, a standard 3×3 convolution, and a 1×1 output convolution, with LeakyReLU after the first two layers. It produces one tolerance value per pixel. Each level has its own adapter, correction network, and tolerance head; within that level, the same modules process TFP and TFI separately. The implementation computes the fusion logit directly as $\tau^l_k - U^l_k/2$; the exponential expression in Eq. (15) explains its confidence interpretation.

The final decoder f_{dec} uses three residual upsampling blocks, additive skips from $\bar{E}$, and residual refinement after each skip sum, followed by three residual blocks and a 3×3 image output convolution.

B.4 Input Construction and Training

Each input contains $T = 45$ binary spike frames, divided into $N = 5$ overlapping blocks of $R = 13$ frames with block-center stride 8. The photoelectric conversion rate is $\eta = 0.6$ for TFP and TFI. Training uses random rotations by multiples of 90° and horizontal flips with probability 0.5, applied consistently to the spikes, starting images, and target.

Flow times follow a logit-normal distribution with location -1, scale 1, and bounds $[0, 0.99]$. Sampled-time states and $t = 0$ starting images are evaluated in one batch with the targets in Eqs. (6) and (10). Training and inference both form fusion candidates from the $t = 0$ outputs using one Euler step; inference evaluates FM only at $t = 0$.

We train all modules jointly from scratch for 500 epochs with Adam, batch size 4, and an initial learning rate of 3×10^{-4}. The learning rate remains constant for the first 300 epochs, then follows cosine decay to 10^{-6} over the remaining 200 epochs. The loss weights in Eq. (20) are $(\lambda_{rec}, \lambda_{base}, \lambda_{fm}, \lambda_{unc}) = (1, 0.1, 0.1, 0.01)$.

C Fusion Ablation Definitions

All three source-assisted variants in Table 4 retain the base feature extractor, both FM sources, proposals P^l_k, and reconstruction decoder, with the same training and evaluation protocol. The alternatives change how these features are fused.

Feature-attention fusion. A gate directly predicts a spatial logit from the base and each source proposal:

$$a_k^l = f_{gate}^l\big([\mathrm{LN}(E^l), \mathrm{LN}(P_k^l)]\big)\,. \tag{22}$$

The gate has the same architecture as f_{tol}^l, but its logit has no explicit uncertainty adjustment. Replacing e_k^l with a_k^l in Eq. (16) retains the base expert and the residual corrections in Eq. (17). This variant tests content-based selection against the combination of content and predicted uncertainty.

Concatenation fusion. This variant combines base, TFP, and TFI features directly:

$$\bar{E}_{cat}^l = f_{cat}^l\big([E^l, P_0^l, P_1^l]\big)\,, \tag{23}$$

where f_{cat}^l is a 1×1 convolution restoring D channels before decoding. It uses neither predicted mixture weights nor uncertainty-guided gating.

D Additional Experimental Results

This section extends the main-paper evaluation with complete synthetic-data visualizations, scene-wise no-reference scores, and the full qualitative comparison on real spike data.

D.1 Complete Synthetic-Data Visualization

Figure 7 extends Fig. 4 to three selected synthetic scenes and all methods in Table 1. For each scene, the first column shows the full GT with the evaluated region and its enlarged crop. The remaining columns pair reconstruction crops with absolute-error maps, computed as pixelwise absolute differences from GT. A common 0–10 gray-level display range allows direct comparison across methods and scenes.

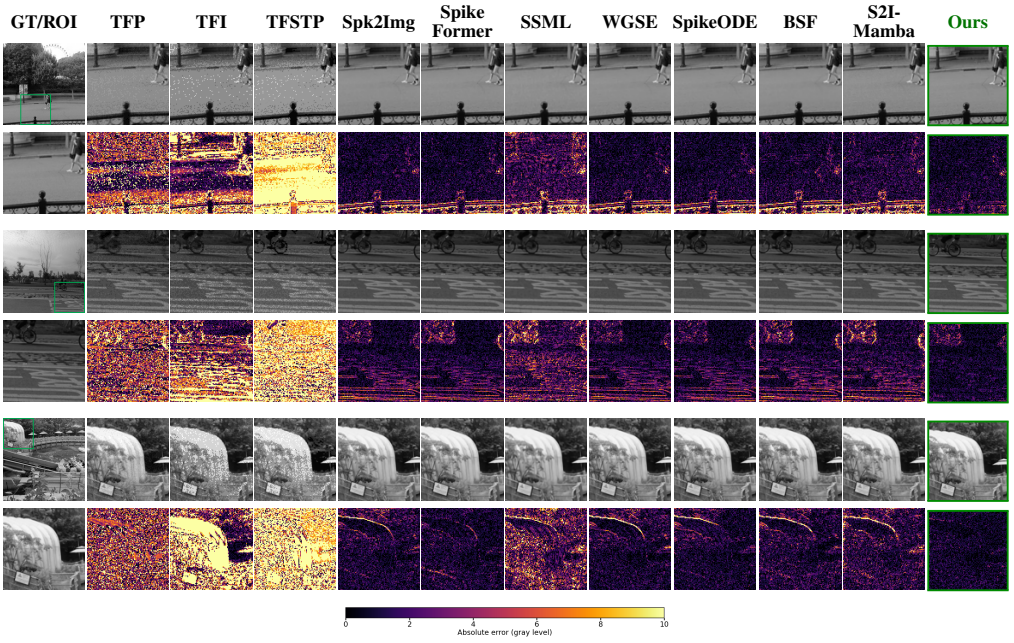

Figure 7: Complete three-scene reconstruction and absolute-error comparison for all methods reported in Table 1. For each scene, the first column shows the full ground truth with the evaluated region and the corresponding ground-truth crop. The remaining columns show reconstructed ROIs in the upper row and absolute-error maps showing the pixelwise absolute difference from GT in the lower row. All error maps use the same displayed range of 0–10 gray levels, with larger values clipped only for visualization. Darker colors indicate smaller errors, and the Ours column is outlined in green.

TFP and TFI exhibit larger residuals: temporal averaging can blur changing structures, while interval-based estimation is sensitive to spike-timing fluctuations. Learning-based methods substantially reduce these errors, although thin boundaries, repeated patterns, and textured regions remain challenging. LTGF produces darker error maps in many of these regions, including the additional third scene. These local improvements agree with its highest PSNR and SSIM and lowest LPIPS in Table 1, supporting gains in both fidelity and perceptual similarity. The results are consistent with selectively transferring useful FM corrections to the base features.

D.2 Scene-Wise Real-Data Quantitative Results

Tables 6–8 provide the HyperIQA, MUSIQ, and MANIQA scores for the six real scenes underlying Table 2. The averages are retained for direct comparison. Rot.-W and Rot.-B denote *Rotation-White* and *Rotation-Black*; bold and underlined values mark the best and second-best results in each column. The compared methods are described in Section 4.2.

LTGF ranks first or second in 16 of the 18 scene–metric combinations. It leads HyperIQA on *Car*, *Rotation-White*, and *Train*; MUSIQ on *Car*, *Rotation-White*, and *Rotation-Black*; and MANIQA on *Car*, *Train*, and *Bridge*. The advantage therefore extends across several scenes, although other methods lead individual comparisons. LTGF's six-scene averages are 0.5510, 68.1292, and 0.6350, respectively, exceeding SpikeODE on all three metrics. These no-reference scores support an overall improvement in assessed image quality on the evaluated real scenes; without GT, they do not directly measure reconstruction fidelity.

D.3 Complete Real-Data Qualitative Comparison

Figure 8 extends Fig. 5 to all compared methods and six real scenes: *Car*, *Rotation-White*, *Rotation-Black*, *Railway*, *Train*, and *Bridge*. Each panel includes the reconstruction and enlarged regions marked by colored boxes. Since GT is unavailable, this comparison is interpreted together with the scene-wise no-reference scores.

The complete comparison exposes the trade-off between smoothing and residual noise. TFP suppresses fluctuations but weakens local boundaries, whereas TFI and TFSTP retain sharper transitions with noisier textures. Learning-based methods improve structure and suppress much of this noise. LTGF preserves fine details in the highlighted wheel spokes in *Car*, character contours in the rotation scenes, rail and sleeper patterns in *Railway*, the diagonal boundary in *Train*, and distant vehicle details in *Bridge*. SpikeODE and LTGF remain visually close in several regions, while LTGF obtains higher average no-reference scores. These examples complement the main-paper evidence for selective generative feature enhancement.

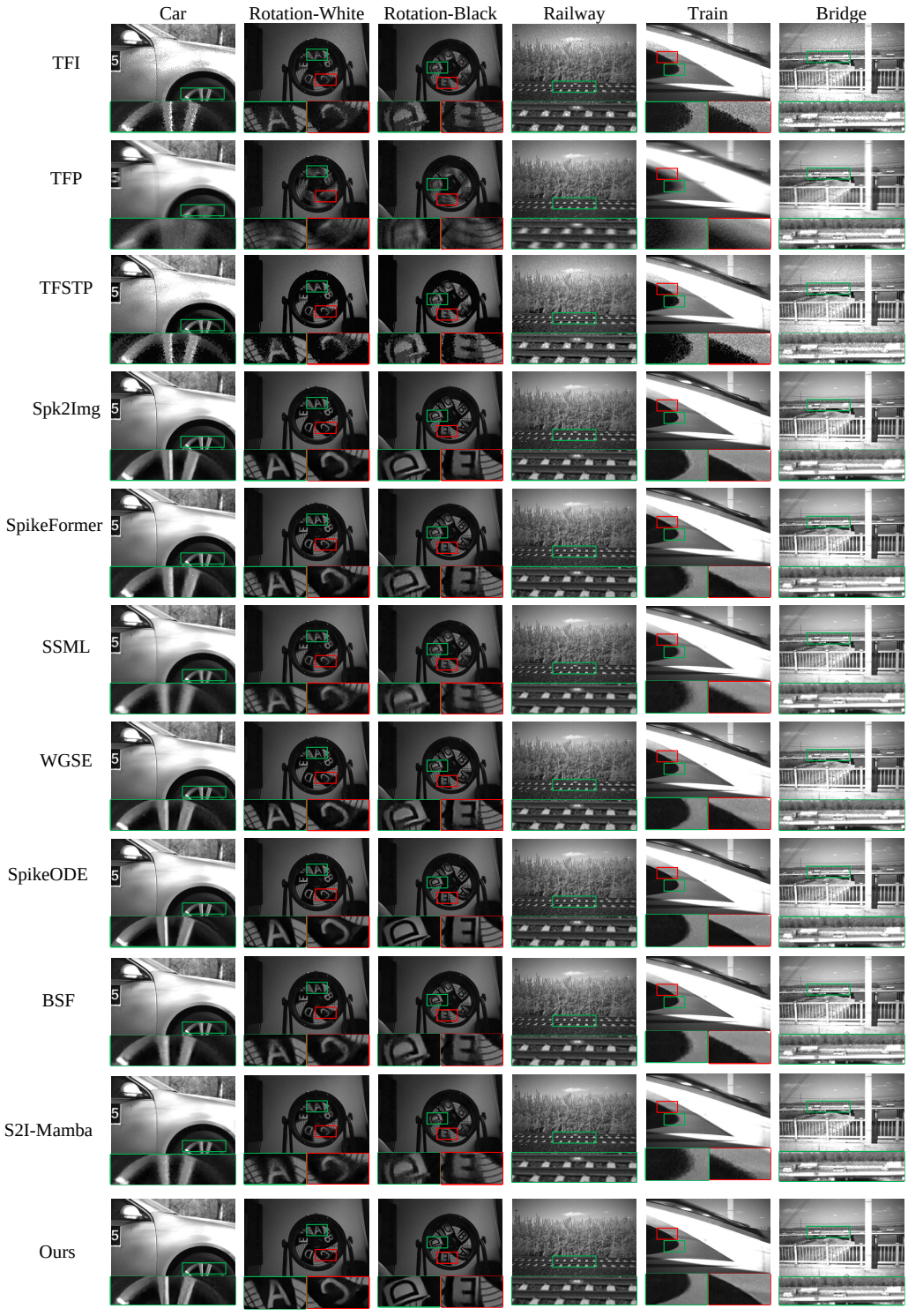

Figure 8: Complete qualitative comparison on all six real spike scenes. Rows show methods and columns show scenes. Colored boxes and enlarged regions highlight fine structures and boundaries.

Table 6: Scene-wise HyperIQA scores on real spike data (higher is better).

Method	Car	Rot.-W	Rot.-B	Railway	Train	Bridge	Avg.
TFI	0.4727	0.5177	0.5160	0.5069	0.3771	0.3959	0.4644
TFP	0.2824	0.4151	0.3893	0.3834	0.2544	0.3245	0.3415
TFSTP	0.4535	0.5329	0.5354	0.5129	0.3578	0.3839	0.4627
Spk2Img	0.5458	0.5416	0.5210	0.5543	0.4677	**0.5396**	0.5283
SpikeFormer	0.5045	0.5272	0.5148	0.5477	0.4261	0.5306	0.5085
SSML	0.4291	0.4614	0.4316	0.4266	0.3429	0.3949	0.4144
WGSE	0.5248	0.5212	0.5023	0.5510	0.4454	0.5324	0.5129
SpikeODE	0.5649	0.5574	**0.5460**	**0.5577**	0.5027	0.5333	0.5437
BSF	0.4897	0.4919	0.4775	0.5453	0.4237	0.5221	0.4917
S2I-Mamba	0.4391	0.4821	0.4634	0.5326	0.3530	0.5006	0.4618
Ours	**0.6042**	**0.5629**	0.5377	0.5574	**0.5063**	0.5375	**0.5510**

Table 7: Scene-wise MUSIQ scores on real spike data (higher is better).

Method	Car	Rot.-W	Rot.-B	Railway	Train	Bridge	Avg.
TFI	66.4970	66.5945	66.0573	67.4326	62.2907	67.1737	66.0076
TFP	53.4317	65.7064	65.1337	65.5198	52.8624	64.8703	61.2541
TFSTP	65.6702	66.5803	66.7286	68.2907	61.8496	66.7512	65.9784
Spk2Img	69.1269	68.1034	67.9368	68.4228	64.5491	68.3306	67.7449
SpikeFormer	68.7071	67.4831	67.1319	68.4171	63.2665	68.3556	67.2269
SSML	66.0600	66.8836	66.3398	67.5029	58.7717	68.3623	65.6534
WGSE	69.0125	67.9593	67.5880	68.3767	63.5447	**68.4300**	67.4852
SpikeODE	69.4340	68.3697	68.3295	**68.5953**	**65.0200**	68.4268	68.0292
BSF	68.0763	67.3751	66.8403	68.2508	62.3872	68.3939	66.8873
S2I-Mamba	66.5740	66.2659	65.9748	67.7563	59.3691	68.1438	65.6807
Ours	**70.1690**	**68.3911**	**68.3653**	68.4315	64.9926	68.4258	**68.1292**

Table 8: Scene-wise MANIQA scores on real spike data (higher is better).

Method	Car	Rot.-W	Rot.-B	Railway	Train	Bridge	Avg.
TFI	0.4091	0.6150	0.6079	0.5880	0.4333	0.4857	0.5232
TFP	0.3792	0.6153	0.5904	0.6265	0.4339	0.5533	0.5331
TFSTP	0.3834	0.5680	0.5557	0.5760	0.4102	0.4843	0.4963
Spk2Img	0.5797	0.6497	0.6363	0.6602	0.5592	0.6283	0.6189
SpikeFormer	0.5506	0.6296	0.6149	0.6585	0.5406	0.6255	0.6033
SSML	0.5298	0.6408	0.6274	**0.6678**	0.5217	0.6162	0.6006
WGSE	0.5727	0.6381	0.6245	0.6608	0.5527	0.6273	0.6127
SpikeODE	0.5847	**0.6610**	**0.6547**	0.6627	0.5808	0.6302	0.6290
BSF	0.5525	0.6240	0.6155	0.6593	0.5464	0.6208	0.6031
S2I-Mamba	0.5155	0.6258	0.6081	0.6545	0.5156	0.6104	0.5883
Ours	**0.6142**	0.6588	0.6458	0.6620	**0.5945**	**0.6346**	**0.6350**

E FUSION WEIGHT DISTRIBUTIONS

Figure 9 shows the test-set distributions of the three fusion weights in Eq. (16) at all four feature levels, from the finest ($l = 0$) to the coarsest ($l = 3$). The base, TFP, and TFI weights are $w_{\varnothing}^{l}$, w_{0}^{l}, and w_{1}^{l}, respectively. The base expert is labeled NULL in the plots because it introduces no residual correction. Each histogram reports the fraction of weight values within each bin.

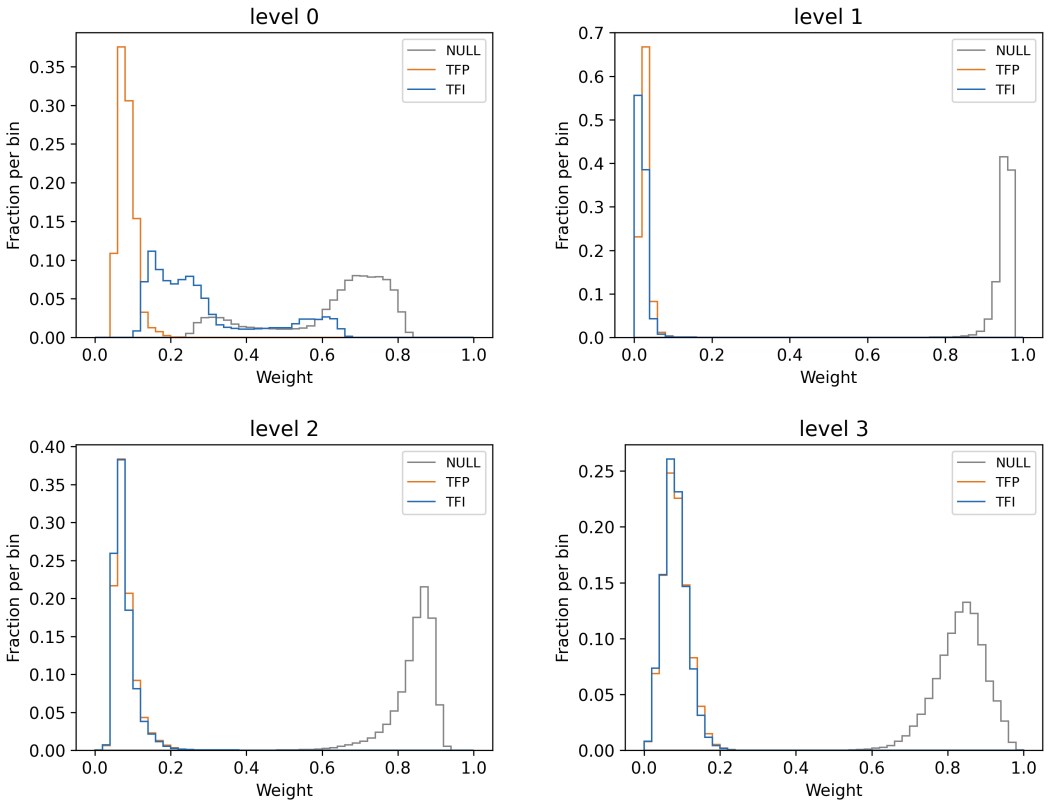

Figure 9: Fusion weight distributions across feature levels. Gray denotes the base expert (NULL), orange TFP, and blue TFI. The horizontal axis gives the weight, and the vertical axis gives the fraction of values per bin. Vertical-axis ranges differ across panels.

Selection across levels. At level 1, the base weight is concentrated near one, while both generative weights remain close to zero, indicating strong suppression of residual corrections at this level. Levels 2 and 3 also favor the base, but admit a broader range of generative weights than level 1. Their TFP and TFI histograms largely overlap. At level 0, the two sources are more clearly separated: TFP remains concentrated at small weights, whereas TFI extends to substantially larger values and the base distribution becomes broader. The finest level therefore assigns larger and more variable weights to TFI corrections. This pattern is consistent with using TFI-derived features to refine local details, while retaining more conservative fusion at the other levels.

Relation to residual enhancement. These weights select residual corrections rather than measure each branch's share of the final image. Equation (17) retains the base feature additively; a large base-expert weight limits the total weight available to FM corrections, and a smaller one does not remove the base feature. The distributions thus show how the model regulates the amount and source of enhancement across levels. Since the logits combine log uncertainty and learned tolerance, they reflect the complete fusion rule. Together with Table 4 and Fig. 6, they support selective use of generative features instead of uniform mixing.

F TEST-SET UNCERTAINTY VALIDATION

The spatial examples in Fig. 6 illustrate how uncertainty relates to reconstruction quality in selected regions. We extend this analysis to all 80 synthetic test samples from 20 scenes (four samples per scene), examining whether U_k^l tracks the error of the corresponding FM candidate before fusion. Each branch generates $\hat{I}_k$ from O_k at $t = 0$, as described in Section 3. We compute its pixelwise squared difference from GT and average-pool the error map to each feature level. TFP and TFI are evaluated separately at levels $l = 0, 1, 2, 3$, from finest to coarsest.

F.1 UNCERTAINTY–ERROR RELATIONSHIP

For each branch and level, we pool all spatial locations across the complete test set, sort them by predicted uncertainty, and divide them into ten equal-count bins, from the lowest 0–10% to the highest 90–100%. Figure 10 reports the mean candidate squared error over the locations in each bin. We estimate pointwise 95% intervals using 2,000 scene-cluster bootstrap resamples: each resample draws 20 scenes with replacement and recomputes the bin means with the original bin assignments fixed. The interval bounds are the 2.5th and 97.5th percentiles across resamples.

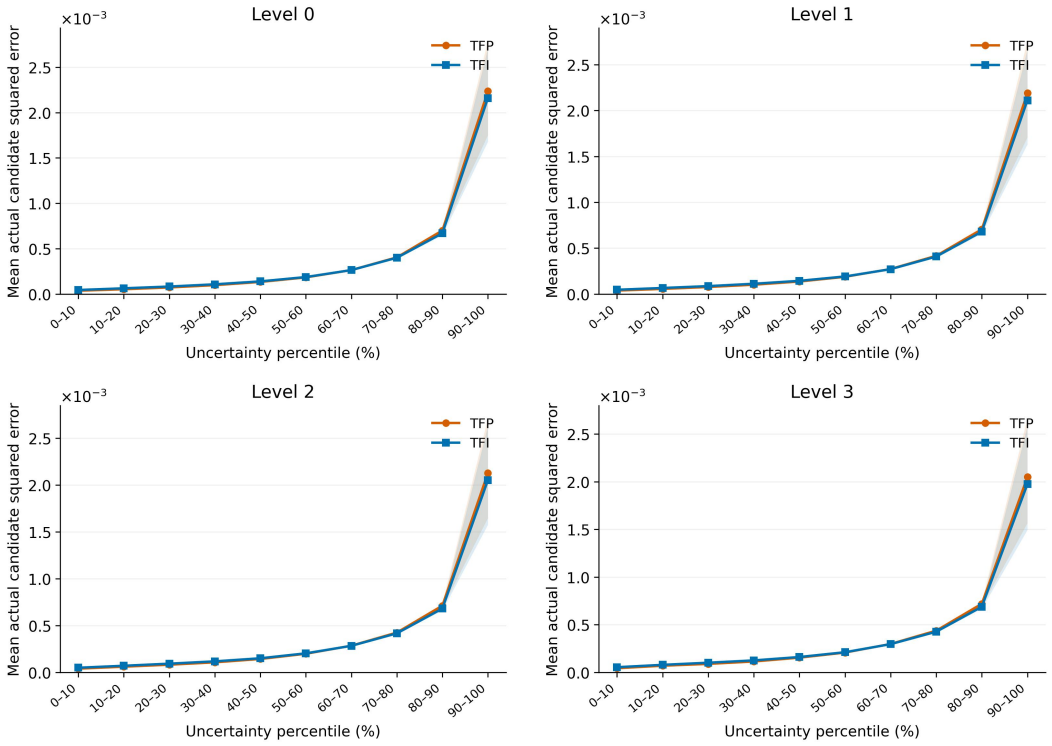

Figure 10: Uncertainty–error relationship at all four feature levels. Orange circles denote TFP and blue squares TFI. Curves show mean candidate squared error in ten global equal-count uncertainty bins per branch and level. Shaded bands denote pointwise 95% scene-cluster bootstrap intervals from 2,000 resamples with fixed bin assignments.

For both branches, mean candidate error increases monotonically with uncertainty at every level. The highest-uncertainty decile shows a particularly sharp increase, while the low-uncertainty bins have substantially smaller errors. Thus, the observed relationship extends beyond the examples in Fig. 6: predicted uncertainty separates candidate regions with different empirical error levels across the test set. Its consistency from the finest to the coarsest level supports using uncertainty as a reliability cue throughout feature fusion. Because the branches are binned separately, their similar curves do not imply that they assign high uncertainty to the same spatial locations.

F.2 Predicted and Empirical Error Magnitudes

The ordering above does not by itself establish agreement in error magnitude. Figure 11 therefore compares the mean predicted squared-error scale $\exp(U_k^l)$ with mean actual candidate squared error within each uncertainty bin. All plotted errors and predicted error scales are computed in the model's native normalized intensity space. Both axes use logarithmic scales, and the dashed diagonal indicates equality. The vertical-axis label E denotes empirical candidate squared error.

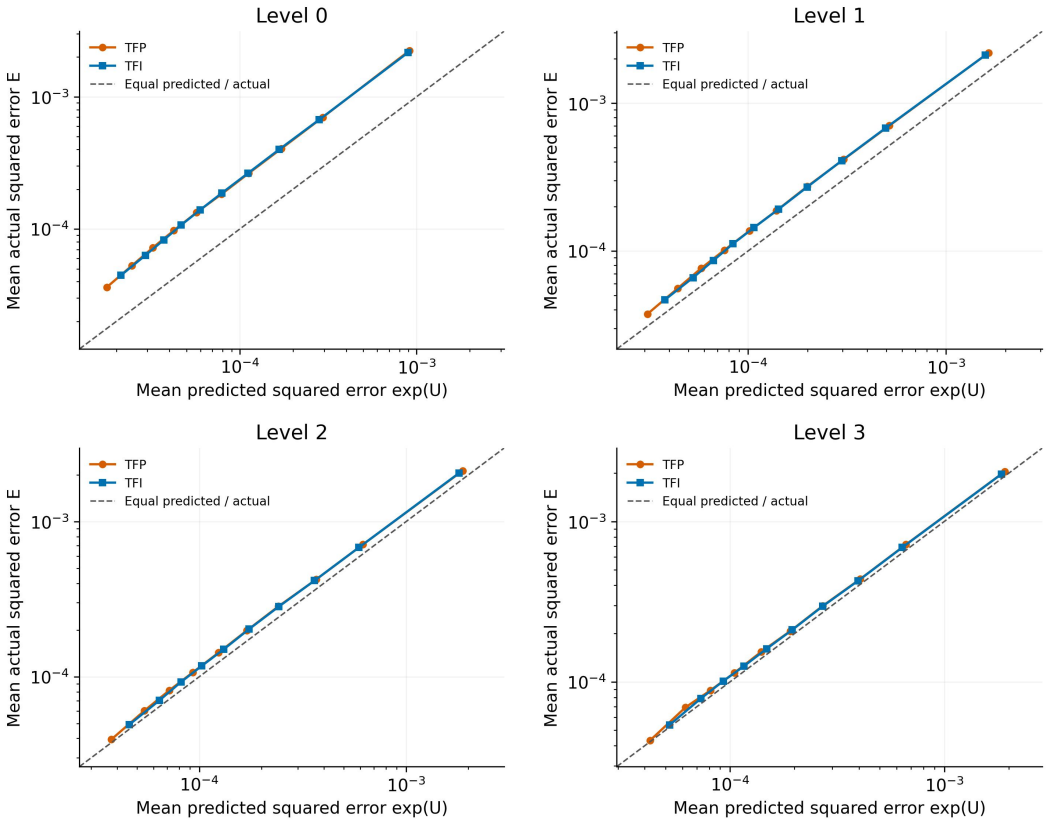

Figure 11: Predicted and empirical candidate error magnitudes at all four levels. Each point compares bin-averaged $\exp(U_k^l)$ with actual squared error. Orange circles denote TFP and blue squares TFI. The dashed line marks equal predicted and actual values; points above it indicate underestimation. Both axes are logarithmic.

Predicted and empirical error magnitudes increase together for both sources, but their agreement varies across levels. Level 0 shows the clearest underestimation, with points systematically above the equality line; Level 1 also retains an upward offset. Levels 2 and 3 lie closer to the line, although a smaller bias remains. The estimates therefore track the empirical error magnitude while exhibiting level-dependent bias. These results support their interpretation as learned error-scale indicators and do not establish exact probabilistic calibration.

Relation to uncertainty-guided fusion. The percentile curves validate uncertainty as a cue for candidate reliability, while the magnitude comparison identifies its limitations. In Eq. (15), this cue is combined with the content-adaptive tolerance τ_k^l, which learns to account for the mismatch between uncertainty and actual fusion benefit. The present plots assess candidate error; the effect on final reconstruction is evaluated by the fusion ablation in Table 4. Together with the weight distributions in Appendix E, these results describe both the reliability information available to fusion and how the model regulates candidate contributions.