ADAPTIVE REUSE OF RESIDUAL POTENTIALS IN LOW-TIMESTEP SPIKING DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models generate high-quality samples through many denoising steps, requiring repeated forward passes of large denoising networks and leading to high inference cost. Spiking neural networks (SNNs) offer a promising way to reduce this cost through sparse spike-based computation, but few neuronal timesteps provide only a coarse approximation of activations, causing errors that can propagate to later denoising steps. We observe that residual potentials retain activation information not yet expressed through spikes. Because consecutive denoising steps operate on successive sample states, this information can remain useful across neighboring steps; however, changes in the sample state and diffusion-time conditioning make full inheritance unreliable. Based on this insight, we propose Residual-Adaptive Spiking Diffusion (RASD), which combines selective residual-potential inheritance with residual-informed threshold adaptation. The former uses a calibrated schedule to control how much residual potential is inherited across denoising steps, while the latter forms historical range observations from residual potentials and spike outputs and uses them to predict the current activation range for threshold adaptation. Experiments on DDPM, Improved-DDPM, and DiT show consistent improvements in low-timestep generation quality with only $T = 4$ or 8 neuronal timesteps, with approximately 65% energy savings.

1 INTRODUCTION

Diffusion models generate high-quality samples through many forward passes of a denoising network, leading to high inference cost. SNNs offer a promising way to reduce this cost through sparse spike-based computation, especially with only a few neuronal timesteps. Since the final sample is obtained through many successive denoising updates rather than a single prediction, it is tempting to assume that a modest approximation at each step can be tolerated, making low-timestep spiking inference particularly appealing.

This intuition, however, becomes less reliable when only a few neuronal timesteps are used. Within limited steps, spike outputs provide only a coarse approximation of the activations, causing errors at each denoising step. These errors can then propagate over to later diffusion steps because each denoising output affects the next sample state. Meanwhile, activation information not yet expressed through spikes remains in the residual potential after the final neuronal timestep. Therefore, resetting potential after each denoising step discards information that could otherwise be used in later steps.

Our key insight is that consecutive denoising steps should not be viewed as independent SNN inferences, because each step operates on the sample state produced by the previous one. This dependence makes the residual from one step useful at the next: it retains activation information not yet expressed through spikes and provides finer-grained information beyond the spike output. However, neighboring denoising steps are similar but not identical, as both the sample state and diffusion-time conditioning change over time. Residual inheritance therefore needs to account for these changes, while the finer-grained information in the residual can also help track the changing activation range.

Based on this insight, we propose Residual-Adaptive Spiking Diffusion (RASD), a framework with selective residual-potential inheritance and residual-informed threshold adaptation. The former uses a calibrated schedule to control how much residual potential is inherited across denoising steps, preserving useful information as local responses change across denoising steps. The latter uses residual

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.

potentials and spike outputs to form historical range observations, from which the current activation range is predicted for threshold adaptation. Together, the two mechanisms improve activation approximation across denoising steps with only a few neuronal timesteps.

We evaluate RASD on DDPM and Improved-DDPM on CIFAR-10 and DiT on ImageNet with as few as 4 neuronal timesteps per denoising step. RASD consistently improves low-timestep generation quality across different diffusion architectures, while requiring only 35% arithmetic energy compared to ANNs. Ablation studies further demonstrate the complementary effects of selective inheritance and threshold adaptation.

Our contributions are summarized as follows:

1. We show that residual potentials retain useful activation information across denoising steps, while changing local responses make full inheritance unreliable, motivating selective residual-potential inheritance.
2. We find that fixed thresholds become mismatched as the activation range changes along the sampling trajectory, and introduce residual-informed threshold adaptation to predict the current range from historical observations and adapt neuronal thresholds accordingly.
3. We validate RASD across different diffusion architectures, demonstrating strong low-timestep generation quality with substantial computational savings.

2 RELATED WORK

2.1 LOW-BIT QUANTIZATION FOR DIFFUSION MODELS

Diffusion quantization is challenging because activation statistics vary across timesteps and errors accumulate during sampling. Early post-training methods use multi-timestep calibration and structure-aware quantization (Shang et al., 2023; Li et al., 2023a). Later methods introduce timestep-conditioned quantizers, temporal-feature preservation, and informative calibration (So et al., 2023; Huang et al., 2024a; Wang et al., 2024), or mitigate accumulated error through noise correction, progressive calibration, timestep-aware correction, and cross-timestep distillation (He et al., 2023; Tang et al., 2025; Yao et al., 2025; Li & Du, 2025). Quantization-awareness training further improves low-bit optimization and efficient fine-tuning (Li et al., 2023b; He et al., 2024).

Recent work extends to sub-2-bit or binary models (Sui et al., 2024; Zheng et al., 2025), image/video DiTs (Wu et al., 2024; Zhao et al., 2025; Chen et al., 2025), and diffusion-specific outlier handling (Lee et al., 2025). Relatedly, Zheng et al. (2024) exploit temporal correlation in BiDM through cross-timestep feature connections. In contrast, we selectively inherit residual potentials across denoising steps.

2.2 ANN-TO-SNN CONVERSION AND SPIKING DIFFUSION

Few-step ANN-to-SNN conversion suffers from coarse spike-based activation approximation. Classical methods calibrate thresholds or parameters (Cao et al., 2015; Diehl et al., 2015; Rueckauer et al., 2017; Li et al., 2021), while later work redesigns neurons, activations, or coding schemes for low latency (Deng & Gu, 2021; Wang et al., 2022; Li & Zeng, 2022; Bu et al., 2022; Jiang et al., 2023; Huang et al., 2025). Transformer conversion further addresses nonlinear operators and attention (Jiang et al., 2024; You et al., 2024; Huang et al., 2024b), while SpikeLLM extends saliency-aware spiking to LLMs (Xing et al., 2025).

Residual potentials retain activation information not expressed through spikes. Prior work preserves such information through soft reset, uses it for conversion-error correction, or calibrates initial potentials to correct spike-count offsets (Han et al., 2020; Hao et al., 2023a;b).

Spiking diffusion has explored U-Nets and temporal mechanisms (Cao et al., 2024; 2025), discrete-latent or synaptic-current formulations (Liu et al., 2024; Watanabe et al., 2023), and spiking Transformer denoisers (Yang et al., 2026). Most closely related, Fang et al. (2026) recycle residual potentials across denoising steps via error feedback. We instead adapt residual inheritance to evolving denoising stage and network location, and further use residual information for threshold adaptation.

3 PRELIMINARIES

3.1 DIFFUSION MODELS

Diffusion models define a forward noising process and generate samples by iteratively reversing it(Ho et al., 2020). Given a clean sample x_0, the noisy state at diffusion step k is

$$x_k = \sqrt{\bar{\alpha}_k} x_0 + \sqrt{1-\bar{\alpha}_k}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{1}$$

where $\alpha_k = 1 - \beta_k$ and $\bar{\alpha}_k = \prod_{j=1}^{k} \alpha_j$. We characterize the denoising stage by the log-SNR

$$\lambda_k = \log \frac{\bar{\alpha}_k}{1-\bar{\alpha}_k}. \tag{2}$$

During sampling, denoising proceeds from $k = N$ to 1. At each step, a denoising network $\epsilon_\theta(x_k, k)$ predicts the noise used to construct the next state. For example, in DDPM sampling,

$$\begin{aligned} \mu_\theta(x_k, k) &= \frac{1}{\sqrt{\alpha_k}} \left(x_k - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}} \epsilon_\theta(x_k, k) \right), \\ x_{k-1} &= \mu_\theta(x_k, k) + \sigma_k z, \qquad z \sim \mathcal{N}(0, I). \end{aligned} \tag{3}$$

Because each denoising prediction determines the input to subsequent steps, approximation errors can propagate along the sampling trajectory. Meanwhile, the changing noise level shifts the denoiser's operating point across steps.

3.2 ANN-TO-SNN CONVERSION AND RESIDUAL POTENTIAL

Within each denoising step, we use $t = 1, \ldots, T$ to index neuronal timesteps. For clarity, we omit the denoising-step and conversion-site indices in this subsection. At a fixed conversion site, a soft-reset IF neuron follows

$$v[t] = v[t-1] + i[t] - V s[t], \tag{4}$$

where $s[t] \in \{0, 1\}$ denotes the binary spike and V denotes the firing threshold.

Let

$$a = \sum_{t=1}^{T} i[t], \qquad y = V \sum_{t=1}^{T} s[t], \qquad v_0 = v[0], \qquad r = v[T]. \tag{5}$$

Summing the neuronal dynamics over the finite neuronal steps provides potential conservation

$$a + v_0 = y + r. \tag{6}$$

With zero initialization, $a = y + r$, or equivalently $r = a - y$. Thus, y represents the portion of the accumulated input expressed through spikes, while r retains the information left unexpressed within the finite neuronal steps. Here, a denotes the actual accumulated input received by the SNN and need not exactly equal the corresponding ANN activation.

4 SELECTIVE RESIDUAL-POTENTIAL INHERITANCE

Residual potentials retain activation information not expressed through spikes within the available neuronal timesteps. Because neighboring denoising steps process successive sample states, this information can be inherited across steps, but changes in the sample state and diffusion-time conditioning can make full inheritance unreliable (Fig. 1). We first analyze the compensation provided by full inheritance and then motivate selective inheritance.

4.1 CROSS-STEP RESIDUAL COMPENSATION

We define the *denoising operating point* by the current sample state and diffusion-time conditioning. Because neighboring steps process successive sample states, their operating points are typically similar. We first analyze the compensation provided by full inheritance.

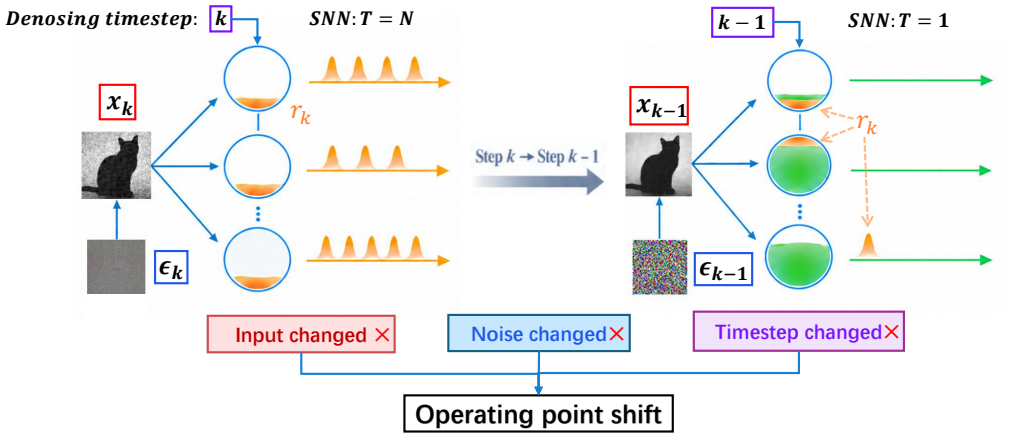


Figure 1: Residual inheritance across denoising steps with changing operating points.

For denoising step k, let a_k, y_k, and $r_k = v_k[T]$ denote the accumulated SNN input, spike-decoded output, and final residual potential, respectively. Since reverse sampling proceeds from $k = N$ to 1, step $k+1$ immediately precedes step k. Full inheritance, as in TA-SpikingDiT (Fang et al., 2026), initializes step k with the residual from step $k+1$:

$$v_k[0] = r_{k+1}, \qquad r_{N+1} = 0. \tag{7}$$

By potential conservation in Eq. 6, $a_k + v_k[0] = y_k + r_k$. Defining the representation error as $e_k \triangleq a_k - y_k$ and substituting Eq. 7 gives

$$e_k = r_k - r_{k+1}. \tag{8}$$

Without inheritance, $e_k = r_k$, and the residual is discarded after each step. Under full inheritance, Eq. 8 telescopes to

$$\sum_{j=k}^{N} e_j = r_k. \tag{9}$$

Thus, the step-wise errors telescope to the remaining residual r_k across steps N to k.

The relation above is exact at the current network location, while its downstream effect depends on the local network response. Let J_k denote the local linearized mapping from a representation perturbation at the considered network location to its downstream response at denoising step k. We use $\sum_{k=1}^{N} J_k e_k$ to characterize the accumulated first-order perturbation along the sampling trajectory.

Proposition 1 (Cross-step residual compensation)**.** *Under potential conservation, full residual inheritance, and local linearization, the accumulated first-order downstream error satisfies*

$$\sum_{k=1}^{N} J_k e_k = J_1 r_1 + \sum_{k=2}^{N} (J_k - J_{k-1}) r_k. \tag{10}$$

The proof is provided in Appendix A.1. When neighboring steps have similar local responses, $J_k - J_{k-1}$ is small, so Eq. 10 approaches the final residual term $J_1 r_1$. As the local response varies across steps, the second term becomes non-negligible, making full inheritance less reliable and motivating selective inheritance.

4.2 Selective Inheritance Scheduling

Eq. 10 shows that full inheritance becomes less reliable as the local response changes across denoising steps. Instead of estimating J_k online, we calibrate the inheritance coefficient offline as a function of denoising stage and network location using representative trajectories.

Using the log-SNR λ_k defined in Eq.2, we introduce an inheritance coefficient $\gamma(\lambda_k; \Gamma)$ to scale the residual inherited from step $k+1$:

$$v_k[0] = \gamma(\lambda_k; \Gamma)\, r_{k+1}, \qquad 0 \le \gamma(\lambda_k; \Gamma) \le 1, \tag{11}$$

where Γ denotes the calibrated inheritance parameters. In practice, γ is calibrated independently for each channel and network location, with their indices absorbed into Γ. By potential conservation, the representation error becomes $e_k = r_k - \gamma(\lambda_k; \Gamma) r_{k+1}$, where $\gamma = 0$ and $\gamma = 1$ recover reset and full inheritance, respectively.

We calibrate γ against the ANN teacher at a small set of log-SNR control points $\{\lambda_i^{\mathrm{c}}\}_{i=1}^{M}$. At each point, we choose γ from a candidate set $\mathcal{G}$ by minimizing teacher-consistency loss:

$$\gamma_i^{\mathrm{c}} = \arg\min_{\gamma \in \mathcal{G}} \frac{1}{|\mathcal{C}_i|} \sum_{\zeta \in \mathcal{C}_i} \ell_{\mathrm{TC}}(\zeta; \gamma), \tag{12}$$

where $\mathcal{C}_i$ contains the calibration samples at control point i, and ℓ_{TC}, typically normalized MSE loss, measures the discrepancy between the SNN and ANN responses at the same network location.

To avoid calibrating every denoising step, we linearly interpolate γ between adjacent log-SNR control points. For $\lambda_i^{\mathrm{c}} \le \lambda_k \le \lambda_{i+1}^{\mathrm{c}}$,

$$\gamma(\lambda_k; \Gamma) = (1 - w_k)\gamma_i^{\mathrm{c}} + w_k \gamma_{i+1}^{\mathrm{c}}, \qquad w_k = \frac{\lambda_k - \lambda_i^{\mathrm{c}}}{\lambda_{i+1}^{\mathrm{c}} - \lambda_i^{\mathrm{c}}}. \tag{13}$$

This yields the inheritance schedule from the calibrated control points.

After calibration, the inheritance schedule is precomputed for all denoising steps and network locations and reused across samples, so inference requires only coefficient lookup and channel-wise residual scaling. The schedule scales only the residual used to initialize the next denoising step; the original residual r_{k+1} is retained for the residual-informed threshold adaptation introduced next.

5 Residual-Informed Threshold Adaptation

Selective inheritance controls how much residual potential is inherited across denoising steps, but does not address changes in the activation range. A fixed threshold may therefore no longer match the activations at different denoising steps. We use residual information from previous steps to estimate the current activation range and adapt the threshold accordingly.

With only a few neuronal timesteps, spike outputs provide only a coarse estimate of the activation range, while the residual potential provides finer-grained activation information beyond the spike output. We use this information to recover the accumulated input of completed denoising steps and form historical range observations, which are then used to predict the activation range of the current step, as shown in Fig. 2.

5.1 Historical Input Observation

For a completed denoising step k, the spike-decoded output y_k, final residual r_k, and initial potential $v_k[0]$ are available. By potential conservation in Eq. 6, the accumulated SNN input can be exactly recovered as

$$\widetilde{a}_k = y_k + r_k - v_k[0] = a_k. \tag{14}$$

Subtracting the initial potential removes the residual inherited from the previous step, leaving the input accumulated during step k. Since the actual initial potential $v_k[0]$ is explicitly removed, the recovery remains exact under reset, full inheritance, and selective inheritance. It uses only quantities available after step k and requires no additional network computation.

At some network locations, diffusion-time conditioning applies a known channel-wise affine modulation:

$$a_k = \eta_k \odot z_k + \delta_k, \tag{15}$$

where η_k and δ_k are the channel-wise scale and shift determined by the diffusion-time conditioning. To compare activation ranges across denoising steps, we remove this known modulation when η_k is element-wise invertible:

$$\widetilde{z}_k = (\widetilde{a}_k - \delta_k) \odot \eta_k^{-1}. \tag{16}$$

This removes only the known diffusion-time modulation; other changes across denoising steps remain in $\widetilde{z}_k$. When the affine modulation cannot be stably inverted, we use $\widetilde{a}_k$ directly for estimation.

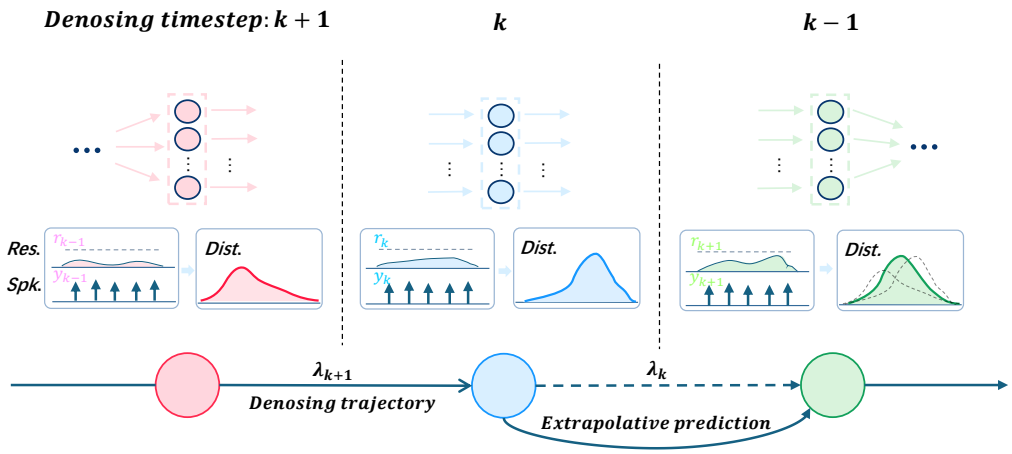


Figure 2: Residual-informed threshold adaptation using historical range prediction.

For each completed denoising step, we summarize the recovered activation by channel-wise lower and upper quantiles:

$$q_k^- = \mathrm{Quantile}_{p^-}(u_k), \qquad q_k^+ = \mathrm{Quantile}_{p^+}(u_k), \qquad h_k = [q_k^-, q_k^+], \tag{17}$$

where the quantiles are computed over non-channel dimensions, with $u_k = \widetilde{z}_k$ when the affine modulation is removed and $u_k = \widetilde{a}_k$ otherwise. h_k is the range observation for step k.

5.2 Causal Range Prediction and Threshold Adaptation

Once two previous range observations are available, we predict the current range by linear extrapolation in log-SNR:

$$\widehat{h}_k = h_{k+1} + \rho_k \frac{\lambda_k - \lambda_{k+1}}{\lambda_{k+1} - \lambda_{k+2}} \left(h_{k+1} - h_{k+2}\right), \qquad 0 \le \rho_k \le 1. \tag{18}$$

The log-SNR ratio accounts for unequal spacing between denoising steps. The damping factor ρ_k controls the extrapolation strength: $\rho_k = 0$ reuses the latest observation, while $\rho_k = 1$ gives full two-point extrapolation.

The predicted $\widehat{h}_k$ is used only to set the threshold for step k and is not stored as a range observation. After step k completes, h_k is recomputed from y_k, r_k, and $v_k[0]$ through Eq. 14, preventing prediction errors from accumulating across denoising steps.

Let $\widehat{h}_k = [\widehat{q_k^-}, \widehat{q_k^+}]$. When the affine modulation is removed before range prediction, we map the predicted quantiles back to the activation space using the known modulation of step k:

$$b_k^{(1)} = \eta_k \odot \widehat{q_k^-} + \delta_k, \qquad b_k^{(2)} = \eta_k \odot \widehat{q_k^+} + \delta_k, \qquad \widehat{R}_k = \max\left(\left|b_k^{(1)}\right|, \left|b_k^{(2)}\right|\right). \tag{19}$$

Taking the larger absolute endpoint gives the predicted symmetric range and remains valid for negative η_k, which only reverses the endpoint ordering. When prediction is performed directly in the activation space, we use $\widehat{R}_k = \max(|\widehat{q_k^-}|, |\widehat{q_k^+}|)$.

To avoid abrupt changes caused by extrapolation, we constrain the predicted range within a fixed ratio of the latest observed range, and denote the constrained resulting range by R_k. For a T-timestep spike representation, the threshold of step k is set as

$$V_k = \frac{R_k}{T}. \tag{20}$$

For the first executed step $k = N$, we use an offline-calibrated range. For $k = N - 1$, only one range observation is available, so we reuse the latest observation. Eq. 18 is applied from $k = N - 2$ onward; additional boundary cases are described in the appendix.

In the joint configuration, selective inheritance sets $v_k[0] = \gamma(\lambda_k; \Gamma) r_{k+1}$, while threshold adaptation sets V_k. After step k completes, the resulting residual r_k is retained for inheritance at step $k - 1$ and, together with y_k and $v_k[0]$, is used to compute the new range observation h_k.

6 EXPERIMENTS

We evaluate our method on low-timestep ANN-to-SNN conversion for diffusion models. We report the overall generation performance and examine the computational efficiency.

6.1 EXPERIMENTAL SETUP

Models and datasets. We train DDPM (Ho et al., 2020) and Improved-DDPM (Nichol & Dhariwal, 2021) on CIFAR-10, and evaluate the publicly available pretrained DiT (Peebles & Xie, 2023) on ImageNet. We focus on low-timestep spiking inference with $T = 4$ and $T = 8$ neuronal timesteps. Further experimental settings are provided in Appendix E.

Baselines. We compare RASD with the original ANN, No-Inheritance, Full-Inheritance, and TA-SpikingDiT (Fang et al., 2026). No-Inheritance resets the initial potential between denoising steps, whereas Full-Inheritance directly inherits the complete residual potential from the preceding step. TA-SpikingDiT is adapted to the corresponding conversion settings for each evaluated architecture. Fixed-ratio inheritance is further evaluated in the ablation study.

Evaluation metrics. We evaluate generation quality using FID, sFID, Inception Score (IS), and Precision (Prec.). All metrics are computed over 10,000 generated samples.

6.2 LOW-TIMESTEP GENERATION AND EFFICIENCY

Generation quality. Table 1 compares RASD with ANN and different low-timestep SNN baselines across DDPM, Improved-DDPM, and DiT. RASD consistently improves generation quality with only a few neuronal timesteps, with particularly clear gains under the more challenging $T = 4$ setting. As T increases to 8, the generation quality further approaches that of the ANN models. The consistent improvements across both convolutional and Transformer-based diffusion models demonstrate that RASD generalizes across different denoising architectures.

Table 1: Generation performance under low neuronal timesteps. ↓ indicates lower is better, while ↑ indicates higher is better.

Model	Method	$T = 4$				$T = 8$			
		FID ↓	**sFID ↓**	**IS ↑**	**Prec. ↑**	**FID ↓**	**sFID ↓**	**IS ↑**	**Prec. ↑**
DDPM	ANN	13.82	6.07	8.66	0.740	13.82	6.07	8.66	0.740
	No-Inheritance	21.90	6.47	8.09	0.707	16.73	6.17	8.35	**0.719**
	Full-Inheritance	23.40	6.95	**8.18**	**0.714**	19.69	6.74	8.34	0.703
	TA-SpikingDiT*	32.90	7.85	7.65	0.692	25.26	7.53	7.32	0.702
	Ours (RASD)	**20.91**	**6.32**	8.14	0.704	**16.46**	**6.03**	**8.52**	0.715
Improved-DDPM	ANN	5.81	4.01	9.57	0.739	5.81	4.01	9.57	0.739
	No-Inheritance	7.83	5.43	9.15	0.726	6.14	4.89	9.41	0.735
	Full-Inheritance	8.04	5.58	9.20	**0.739**	6.26	4.98	9.34	0.724
	TA-SpikingDiT*	7.76	5.60	9.08	0.725	6.11	4.67	9.37	0.730
	Ours (RASD)	**7.52**	**5.10**	**9.22**	0.713	**6.05**	**4.56**	**9.43**	**0.736**
DiT-XL/2-256 (steps=25)	ANN	5.76	22.62	179.12	0.801	5.76	22.62	179.12	0.801
	No-Inheritance	9.03	24.78	165.26	0.724	7.15	23.52	170.18	0.731
	Full-Inheritance	7.17	24.25	166.86	0.750	6.50	23.21	171.34	0.766
	TA-SpikingDiT*	5.94	23.56	172.10	0.782	5.88	22.80	174.55	0.794
	Ours (RASD)	**5.91**	**23.14**	**173.05**	**0.783**	**5.83**	**22.75**	**174.86**	**0.800**

* TA-SpikingDiT is adapted to the corresponding conversion settings of the evaluated architectures, with the MT threshold set to 8.

Interestingly, Full-Inheritance shows a clear metric divergence at $T = 4$: it improves IS and Precision over No-Inheritance, while degrading FID and sFID. This suggests that full inheritance introduces two competing effects. The residual potential retains activation information not expressed

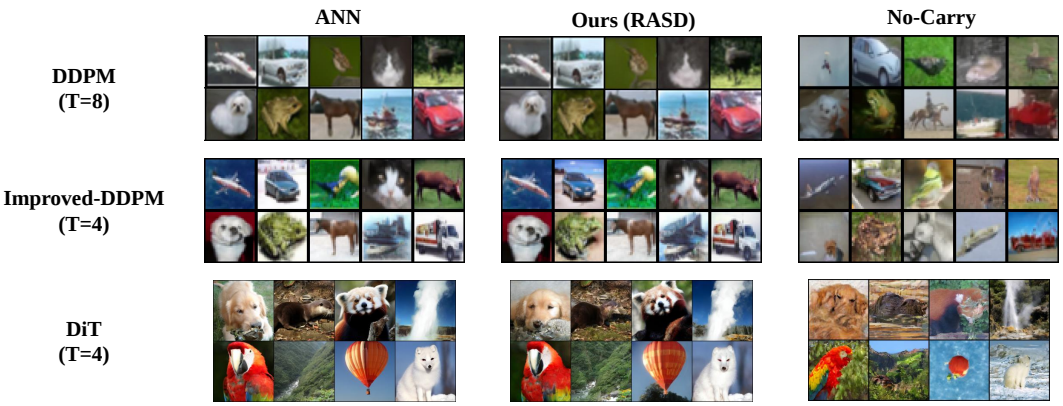


Figure 3: Qualitative comparison of ANN, No-Inheritance, and RASD under low neuronal timesteps across different diffusion architectures.

through spikes and can therefore compensate for coarse low-timestep approximation, while changes in the local response across denoising steps make complete inheritance increasingly mismatched to the current operating point. RASD addresses this trade-off by selectively inheriting residual information, consistent with the analysis in Sec. 4. At $T = 8$, this effect becomes weaker as more activation information is already expressed through spikes.

Computational efficiency. In addition to generation quality, we evaluate the computational cost of low-timestep spiking inference. Following the standard 45-nm energy model in Horowitz (2014), we estimate 4.6 pJ for an MAC and 0.9 pJ for an AC operation. Most converted operations are executed with spike-driven ACs, while a small set of input and attention-related operations remains MAC-based. As shown in Table 2, at $T = 4$, RASD requires only 35.18% and 35.63% of the ANN arithmetic energy on DDPM and Improved-DDPM, respectively.

Table 2: Estimated computational cost and relative arithmetic energy of ANN and RASD.

Model	Config	MACs (G)	ACs (G)	Energy (mJ)	Relative Energy
DDPM	ANN	6.22	0.01	28.621	100%
	RASD ($T = 4$)	0.52	8.53	10.069	**35.18%**
Improved-DDPM	ANN	8.39	0.01	38.603	100%
	RASD ($T = 4$)	0.86	10.89	13.757	**35.63%**

6.3 Mechanistic Analysis of Residual Inheritance

Analysis of Residual Inheritance We further examine the two observations underlying selective residual-potential inheritance: whether residual potentials provide useful compensation along the sampling trajectory, and why full inheritance becomes less reliable as the denoising operating point evolves. For each inheritance strategy, we run its own sampling trajectory and compare the denoising output at each diffusion timestep with the corresponding ANN output.

As shown in Fig. 4(a), Selective-Inheritance yields lower ANN-relative MSE than No-Inheritance across most denoising stages. This indicates that residual potentials retain activation information not expressed through spikes and can compensate for part of the low-timestep approximation error. In contrast, Full-Inheritance produces a larger discrepancy over much of the trajectory. This is consistent with the analysis in Sec. 4.1: as the local response changes across denoising steps, directly inheriting the complete residual becomes less reliable.

The trajectory-level improvement also carries through to the final generated samples. As shown in Fig. 4(b), Selective-Inheritance achieves the lowest final-sample MSE to the ANN reference among the three strategies. The paired comparison in Fig. 4(c) further shows improvements on 44 out of 64 samples over No-Inheritance, with a median MSE reduction of 13.2%. These results show

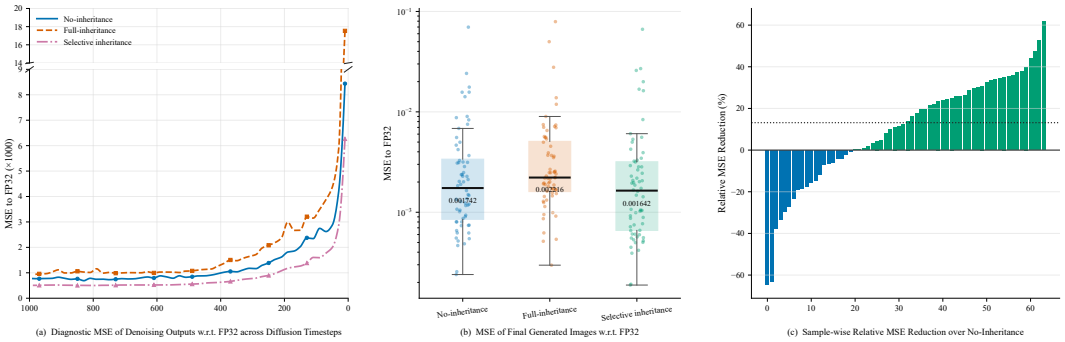


Figure 4: Analysis of residual inheritance. (a) MSE of denoising outputs relative to ANN along the sampling trajectory. (b) MSE of final generated samples relative to ANN. (c) Sample-wise relative MSE reduction of Selective-Inheritance over No-Inheritance.

that selective inheritance reduces the approximation gap along the sampling trajectory and that this benefit persists to the final samples across a majority of trajectories.

Residual inheritance ablation. We further compare selective inheritance with fixed inheritance coefficients in Table 3. All variants use the same threshold-estimation procedure, so the comparison isolates the effect of residual inheritance. Fixed residual scaling provides no consistent improvement over No-Inheritance, while Selective-Inheritance achieves a lower FID. This supports adapting the inherited residual across denoising stages and network locations rather than using a fixed coefficient throughout the sampling trajectory.

Table 3: Comparison of fixed and selective residual inheritance strategies. All variants use the same threshold-estimation procedure.

Method	**No-Inheritance**	$\gamma = 0.25$	$\gamma = 0.50$	$\gamma = 0.75$	**Full-Inheritance**	**Selective-Inheritance**
FID $\downarrow$	7.83	7.85	7.96	7.94	8.04	**7.52**

Additional analyses of residual-error accumulation, historical input Observation, and activation-range prediction are provided in the Appendix F.

7 Conclusion

We proposed RASD for low-timestep spiking diffusion through selective residual-potential inheritance and residual-informed threshold adaptation. At only a few neuronal timesteps, ANN-to-SNN conversion may appear similar to activation quantization. Yet SNNs retain additional information: activation information not expressed through spikes remains in the residual potential. This information can remain useful at the next denoising step, but cannot be inherited without accounting for changes in local responses. Diffusion therefore provides both the need and the opportunity for such reuse: neighboring denoising steps are different enough to make full inheritance and fixed thresholds unreliable, yet similar enough for residual information and recent range observations to remain informative. RASD uses this structure through selective inheritance and threshold adaptation. These results suggest that low-timestep spiking inference can benefit not only from sparse spike-based computation, but also from reusing residual information across successive denoising steps.

Our current inheritance schedule is calibrated from representative trajectories and shared across samples. It captures stable trends across denoising stages and network locations, but the appropriate inheritance coefficient may also depend on the current sample and its local response. Estimating this reliably online remains difficult. A natural next step is therefore to make residual inheritance sample-adaptive. More broadly, it would be interesting to ask whether similar state reuse can benefit other iterative inference processes with strongly correlated successive states.

8 AI ASSISTANCE

AI tools were used solely to improve the clarity, grammar, and readability of the manuscript. All AI-assisted text was reviewed and revised by the authors.

REFERENCES

Tong Bu, Wei Fang, Jianhao Ding, Penglin Dai, Zhaofei Yu, and Tiejun Huang. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7B3IJMM1k_M.

Jiahang Cao, Ziqing Wang, Hanzhong Guo, Hao Cheng, Qiang Zhang, and Renjing Xu. Spiking denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4912–4921, January 2024.

Jiahang Cao, Hanzhong Guo, Ziqing Wang, Deming Zhou, Hao Cheng, Qiang Zhang, and Renjing Xu. Spiking diffusion models. *IEEE Transactions on Artificial Intelligence*, 6(1):132–143, 2025. doi: 10.1109/TAI.2024.3453229.

Yongqiang Cao, Yang Chen, and Deepak Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113(1):54–66, 2015. ISSN 1573-1405. doi: 10.1007/s11263-014-0788-3. URL https://doi.org/10.1007/s11263-014-0788-3.

Lei Chen, Yuan Meng, Chen Tang, Xinzhu Ma, Jingyan Jiang, Xin Wang, Zhi Wang, and Wenwu Zhu. Q-dit: Accurate post-training quantization for diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28306–28315, June 2025.

Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=FZ1oTwcXchK.

Peter U. Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2015. doi: 10.1109/IJCNN.2015.7280696.

Haoran Fang, Tianxing Man, Xingchen Li, Wanli Shi, Jinjie Fang, and Bin Gu. Trajectory-aware spiking dits conversion via membrane potential error-feedback. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=nsd26wlIGP.

Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Zecheng Hao, Tong Bu, Jianhao Ding, Tiejun Huang, and Zhaofei Yu. Reducing ann-snn conversion error through residual membrane potential. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(1):11–21, 2023a. doi: 10.1609/aaai.v37i1.25071.

Zecheng Hao, Jianhao Ding, Tong Bu, Tiejun Huang, and Zhaofei Yu. Bridging the gap between ANNs and SNNs by calibrating offset spikes. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=PFbzoWZyZRX.

Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-training quantization for diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 13237–13249. Curran Associates, Inc., 2023. doi: 10.52202/075280-0580. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/2aab8a76c7e761b66eccaca0927787de-Paper-Conference.pdf.

Yefei He, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. EfficientDM: Efficient quantization-aware fine-tuning of low-bit diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=UmMa3UNDAz.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14. IEEE, 2014. doi: 10.1109/ISSCC.2014.6757323.

Yushi Huang, Ruihao Gong, Jing Liu, Tianlong Chen, and Xianglong Liu. Tfmq-dm: Temporal feature maintenance quantization for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7362–7371, June 2024a.

Zihan Huang, Xinyu Shi, Zecheng Hao, Tong Bu, Jianhao Ding, Zhaofei Yu, and Tiejun Huang. Towards high-performance spiking transformers from ann to snn conversion. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, pp. 10688–10697, New York, NY, USA, 2024b. Association for Computing Machinery. ISBN 9798400706868. doi: 10.1145/3664647.3680620. URL https://doi.org/10.1145/3664647.3680620.

Zihan Huang, Wei Fang, Tong Bu, Peng Xue, Zecheng Hao, Wenxuan Liu, Yuanhong Tang, Zhaofei Yu, and Tiejun Huang. Differential coding for training-free ANN-to-SNN conversion. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25235–25264. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/huang25i.html.

Haiyan Jiang, Srinivas Anumasa, Giulia De Masi, Huan Xiong, and Bin Gu. A unified optimization framework of ANN-SNN conversion: Towards optimal mapping from activation values to firing rates. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14945–14974. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/jiang23a.html.

Yizhou Jiang, Kunlin Hu, Tianren Zhang, Haichuan Gao, Yuqian Liu, Ying Fang, and Feng Chen. Spatio-temporal approximation: A training-free SNN conversion for transformers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=XrunSYwoLr.

Dongyeun Lee, Jiwan Hur, Hyounguk Shon, Jae Young Lee, and Junmo Kim. Dmq: Dissecting outliers of diffusion models for post-training quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18510–18520, October 2025.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17535–17545, October 2023a.

Yang Li and Yi Zeng. Efficient and accurate conversion of spiking neural network with burst spikes. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 2485–2491. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/345. URL https://doi.org/10.24963/ijcai.2022/345.

Yanjing Li, Sheng Xu, Xianbin Cao, Xiao Sun, and Baochang Zhang. Q-dm: An efficient low-bit quantized diffusion model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 76680–76691. Curran Associates, Inc., 2023b. doi: 10.52202/

075280-3350. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/f1ee1cca0721de55bb35cf28ab95e1b4-Paper-Conference.pdf.

Yanxi Li and Chengbin Du. Optimizing Quantized Diffusion Models via Distillation with Cross-Timestep Error Correction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39 (17):18530–18538, 2025. doi: 10.1609/aaai.v39i17.34039. URL https://doi.org/10.1609/aaai.v39i17.34039.

Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ann: Towards efficient, accurate spiking neural networks calibration. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6316–6325. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/li21d.html.

Mingxuan Liu, Jie Gan, Rui Wen, Tao Li, Yongli Chen, and Hong Chen. Spiking-diffusion: Vector quantized discrete diffusion model with spiking neural networks. In *2024 5th International Conference on Computer, Big Data and Artificial Intelligence (ICCBD+AI)*, pp. 627–631, 2024. doi: 10.1109/ICCBD-AI65562.2024.00110.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023. doi: 10.1109/ICCV51070.2023.00387.

Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, Volume 11 - 2017, 2017. ISSN 1662-453X. doi: 10.3389/fnins.2017.00682. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2017.00682.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1972–1981, June 2023.

Junhyuk So, Jungwon Lee, Daehyun Ahn, Hyungjun Kim, and Eunhyeok Park. Temporal dynamic quantization for diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 48686–48698. Curran Associates, Inc., 2023. doi: 10.52202/075280-2113. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/983591c3e9a0dc94a99134b3238bbe52-Paper-Conference.pdf.

Yang Sui, Yanyu Li, Anil Kag, Yerlan Idelbayev, Junli Cao, Ju Hu, Dhritiman Sagar, Bo Yuan, Sergey Tulyakov, and Jian Ren. Bitsfusion: 1.99 bits weight quantization of diffusion model. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 76775–76818. Curran Associates, Inc., 2024. doi: 10.52202/079017-2443. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/8c64bc3f7796d31caa7c3e6b969bf7da-Paper-Conference.pdf.

Siao Tang, Xin Wang, Hong Chen, Chaoyu Guan, Zewen Wu, Yansong Tang, and Wenwu Zhu. Post-training quantization with progressive calibration and activation relaxing for text-to-image diffusion models. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 404–420, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72992-8.

Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Towards accurate post-training quantization for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16026–16035, June 2024.

Yuchen Wang, Malu Zhang, Yi Chen, and Hong Qu. Signed neuron with memory: Towards simple, accurate and high-efficient ann-snn conversion. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 2501–2508. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/347. URL https://doi.org/10.24963/ijcai.2022/347. Main Track.

Ryo Watanabe, Yusuke Mukuta, and Tatsuya Harada. Fully spiking denoising diffusion implicit models, 2023. URL https://arxiv.org/abs/2312.01742.

Junyi Wu, Haoxuan Wang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Ptq4dit: Post-training quantization for diffusion transformers. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 62732–62755. Curran Associates, Inc., 2024. doi: 10.52202/079017-2006. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/72d32f4fe0b7af03732bd227bf1c4a5f-Paper-Conference.pdf.

Xingrun Xing, Boyan Gao, Zheng Liu, David A. Clifton, Shitao Xiao, Wanpeng Zhang, Li Du, Zheng Zhang, Guoqi Li, and Jiajun Zhang. SpikeLLM: Scaling up spiking neural network to large language models via saliency-based spiking. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZadnlOHsHv.

Shu Yang, Hanzhi Ma, Chengting Yu, Aili Wang, Pengqi Fu, and Er-Ping Li. Sdit: Scalable and efficient spiking diffusion transformers for image generation. *Pattern Recognition Letters*, 207:266–273, 2026. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2026.07.003. URL https://www.sciencedirect.com/science/article/pii/S0167865526002333.

Yuzhe Yao, Feng Tian, Jun Chen, Haonan Lin, Guang Dai, Yong Liu, and Jingdong Wang. Timestep-aware correction for quantized diffusion models. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 215–232, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72848-8.

Kang You, Zekai Xu, Chen Nie, Zhijie Deng, Qinghai Guo, Xiang Wang, and Zhezhi He. SpikeZIP-TF: Conversion is all you need for transformer-based SNN. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57367–57383. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/you24b.html.

Tianchen Zhao, Tongcheng Fang, Haofeng Huang, Rui Wan, Widyadewi Soedarmadji, Enshu Liu, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. Vidit-q: Efficient and accurate quantization of diffusion transformers for image and video generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=E1N1oxd63b.

Xingyu Zheng, Xianglong Liu, Yichen Bian, Xudong Ma, Yulun Zhang, Jiakai Wang, Jinyang Guo, and Haotong Qin. Bidm: Pushing the limit of quantization for diffusion models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 39009–39035. Curran Associates, Inc., 2024. doi: 10.52202/079017-1231. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/44b61c5c0ba06d55ab5a1cfb9cfff763-Paper-Conference.pdf.

Xingyu Zheng, Xianglong Liu, Haotong Qin, Xudong Ma, Mingyuan Zhang, Haojie Hao, Jiakai Wang, Zixiang Zhao, Jinyang Guo, and Michele Magno. BinaryDM: Accurate weight binarization for efficient diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=YaeZwhXJ4k.

A Analysis of Residual-Potential Inheritance

A.1 Proof of Full-Inheritance Compensation

We first prove the compensation relation used in Sec. 4.1. Recall that reverse sampling proceeds from $k = N$ to 1, so the denoising step immediately preceding step k is step $k + 1$. Under full residual inheritance,

$$v_k[0] = r_{k+1}, \qquad r_{N+1} = 0. \tag{21}$$

Together with the potential-conservation relation, this gives

$$e_k = a_k - y_k = r_k - v_k[0] = r_k - r_{k+1}, \tag{22}$$

consistent with Eq. equation 8.

Let J_k denote the first-order downstream response associated with the local representation error at denoising step k, as defined in the main text. The accumulated first-order discrepancy is therefore

$$\begin{aligned}\sum_{k=1}^{N} J_k e_k &= \sum_{k=1}^{N} J_k (r_k - r_{k+1}) \\ &= \sum_{k=1}^{N} J_k r_k - \sum_{k=1}^{N} J_k r_{k+1}.\end{aligned} \tag{23}$$

Using $r_{N+1} = 0$ and re-indexing the second term,

$$\sum_{k=1}^{N} J_k r_{k+1} = \sum_{k=2}^{N} J_{k-1} r_k. \tag{24}$$

Hence,

$$\sum_{k=1}^{N} J_k e_k = J_1 r_1 + \sum_{k=2}^{N} (J_k - J_{k-1})\, r_k, \tag{25}$$

which proves Eq. equation 10.

Eq. equation 25 separates the remaining first-order discrepancy into the terminal residual contribution $J_1 r_1$ and the mismatch terms induced by changes in the downstream response between neighboring denoising steps. When neighboring responses are similar, the intermediate residual contributions largely cancel across steps.

A.2 Extension to Selective Inheritance

We next extend the compensation relation to selective inheritance. For compactness, let

$$\gamma_k = \gamma(\lambda_k; \Gamma), \qquad 0 \le \gamma_k \le 1, \tag{26}$$

and let r_k^γ denote the residual potential produced under the corresponding inheritance schedule, with $r_{N+1}^\gamma = 0$. The initial potential at denoising step k is then

$$v_k[0] = \gamma_k r_{k+1}^\gamma. \tag{27}$$

By potential conservation, the corresponding local representation error satisfies

$$e_k^\gamma = r_k^\gamma - \gamma_k r_{k+1}^\gamma. \tag{28}$$

Using the same first-order downstream response notation as in Sec. A.1, the accumulated discrepancy becomes

$$\begin{aligned}\sum_{k=1}^{N} J_k e_k^\gamma &= \sum_{k=1}^{N} J_k \left(r_k^\gamma - \gamma_k r_{k+1}^\gamma\right) \\ &= \sum_{k=1}^{N} J_k r_k^\gamma - \sum_{k=1}^{N} \gamma_k J_k r_{k+1}^\gamma.\end{aligned} \tag{29}$$

Re-indexing the second term and using $r^{\gamma}_{N+1} = 0$ gives

$$\sum_{k=1}^{N} J_k e^{\gamma}_k = J_1 r^{\gamma}_1 + \sum_{k=2}^{N} \left(J_k - \gamma_{k-1} J_{k-1}\right) r^{\gamma}_k. \tag{30}$$

Setting $\gamma_k = 1$ for all k recovers the full-inheritance relation in Eq. equation 25. More generally, selective inheritance replaces the full-inheritance mismatch term $J_k - J_{k-1}$ with $J_k - \gamma_{k-1} J_{k-1}$. Thus, γ_{k-1} directly regulates how strongly the residual produced at denoising step k participates in compensation at the subsequent step $k-1$.

Importantly, r^{γ}_k itself depends on the inheritance schedule, since changing the initial potential also changes the subsequent spike outputs and residual potentials. Eq. equation 30 therefore characterizes how the inheritance coefficient enters the compensation mechanism, but does not provide an analytic optimum for γ_k or imply that weaker inheritance is always preferable. The inheritance schedule is instead determined by offline calibration, as described in Appendix C.1.

B Residual Inheritance under Stochastic Sampling

Appendix A analyzes cross-step residual compensation along a fixed sampling trajectory. We next examine how its endpoint effect varies under stochastic sampling. We consider a controlled branching setting in which the history before a selected stochastic innovation is fixed, all other stochastic innovations are shared, and only the selected innovation is varied. For each realization, all compared variants use the same stochastic innovations and differ only in the residual-inheritance strategy. We refer to the resulting continuations as stochastic branches.

B.1 Endpoint Gain across Stochastic Branches

For each stochastic branch, let x^{ref}, x^{no}, and x^{Γ} denote the final outputs of the reference model, the SNN without residual inheritance, and the SNN using inheritance schedule Γ, respectively. Define

$$\varepsilon = x^{\mathrm{ref}} - x^{\mathrm{no}}, \qquad c = x^{\Gamma} - x^{\mathrm{no}}, \tag{31}$$

where ε is the endpoint discrepancy without inheritance and c is the endpoint correction induced by inheritance. The remaining discrepancy is therefore $\varepsilon - c$.

As an analytical measure of inheritance gain, we consider the reduction in squared endpoint discrepancy:

$$G = \|\varepsilon\|^2 - \|\varepsilon - c\|^2 = 2\langle \varepsilon, c\rangle - \|c\|^2. \tag{32}$$

Thus, a larger endpoint correction does not necessarily yield a larger gain; the correction must also be aligned with the discrepancy it is intended to reduce.

Let $\mathcal{H}$ denote the fixed history before the varied stochastic innovation, and let the expectation below be taken only over this innovation while the remaining stochastic innovations are fixed. Define

$$\bar{\varepsilon} = \mathbb{E}[\varepsilon \mid \mathcal{H}], \qquad \bar{c} = \mathbb{E}[c \mid \mathcal{H}]. \tag{33}$$

Assuming finite second moments, the expected gain satisfies

$$\begin{aligned} \mathbb{E}[G \mid \mathcal{H}] = {} & 2\langle \bar{\varepsilon}, \bar{c}\rangle - \|\bar{c}\|^2 \\ & + 2\mathbb{E}[\langle \varepsilon - \bar{\varepsilon}, c - \bar{c}\rangle \mid \mathcal{H}] \\ & - \mathbb{E}\left[\|c - \bar{c}\|^2 \mid \mathcal{H}\right]. \end{aligned} \tag{34}$$

This follows directly by substituting $\varepsilon = \bar{\varepsilon} + (\varepsilon - \bar{\varepsilon})$ and $c = \bar{c} + (c - \bar{c})$ into Eq. equation 32 and taking the conditional expectation.

Eq. equation 34 separates the effect of stochastic branching into three parts. The first term measures the benefit of the mean correction, the second captures whether branch-dependent changes in the correction follow corresponding changes in the endpoint discrepancy, and the last penalizes branch-dependent variation in the correction itself. Stochastic variation is therefore not inherently detrimental to residual inheritance; its effect depends on whether the resulting correction remains aligned with the discrepancy to be reduced.

B.2 Implications for Branch-Wise Prediction

The branch dependence above also limits what can be inferred from the pre-branch history alone. Consider any predictor $\widehat{G}(\mathcal{H})$ that uses only $\mathcal{H}$ to estimate the final gain. Its conditional mean-squared error satisfies

$$\begin{aligned}\mathbb{E}\Big[(G-\widehat{G}(\mathcal{H}))^2 \mid \mathcal{H}\Big] &= \mathrm{Var}(G \mid \mathcal{H}) + \Big(\mathbb{E}[G \mid \mathcal{H}] - \widehat{G}(\mathcal{H})\Big)^2 \\ &\geq \mathrm{Var}(G \mid \mathcal{H}). \end{aligned} \tag{35}$$

The lower bound is attained by $\widehat{G}(\mathcal{H}) = \mathbb{E}[G \mid \mathcal{H}]$.

In particular, if the same history admits both positive- and negative-gain branches with nonzero probability, then $\mathrm{Var}(G \mid \mathcal{H}) > 0$. Under this controlled branching setting, the pre-branch history alone therefore cannot determine the final gain of every stochastic realization exactly.

This observation does not rule out online adaptation, nor does it imply that an offline inheritance schedule is optimal. It only shows that a predictor restricted to the same pre-branch history need not agree with the eventual branch-wise gain for every stochastic realization. Our method does not rely on such branch-wise prediction. Instead, $\gamma(\lambda_k;\Gamma)$ captures the more stable dependence on denoising stage and network location, while Γ is calibrated over representative trajectories as described in Appendix C.1.

C Calibration and Implementation of Selective Inheritance

C.1 Offline Calibration of the Inheritance Schedule

We parameterize the inheritance schedule $\gamma(\lambda;\Gamma)$ using a small set of control values over log-SNR and network location. For implementation, the converted locations are organized into $G = 6$ network groups, with separate coefficients for the feature branch (F) and conditioning branch (C):

$$\Gamma = \{\gamma_{i,g,b}\}_{i=1,\ldots,M;\, g=1,\ldots,G;\, b\in\{\mathrm{F},\mathrm{C}\}}, \tag{36}$$

where i indexes the log-SNR control points. We use $M = 15$ control points spanning the log-SNR range of the sampling trajectory, resulting in $15 \times 6 \times 2 = 180$ control values.

The control values are calibrated once using representative trajectories and the ANN teacher. At each log-SNR control point λ_i^{c}, the inheritance coefficient for each network group and branch is selected independently from a finite candidate set $\mathcal{G} \subseteq [0,1]$:

$$\gamma_{i,g,b}^{\mathrm{c}} = \arg\min_{\gamma\in\mathcal{G}} \frac{1}{|\mathcal{C}_i|} \sum_{\zeta\in\mathcal{C}_i} \ell_{\mathrm{TC}}(\zeta;\gamma,g,b), \tag{37}$$

where $\mathcal{C}_i$ contains the calibration samples at control point i, and ℓ_{TC} measures the discrepancy between the SNN and ANN responses at the corresponding network location. All other conversion settings are fixed during calibration. We use normalized MSE as the teacher-consistency loss.

At inference time, the coefficient at each denoising step is obtained by linear interpolation between adjacent log-SNR control points. For $\lambda_i^{\mathrm{c}} \leq \lambda_k \leq \lambda_{i+1}^{\mathrm{c}}$,

$$\gamma_{k,g,b} = (1-w_k)\gamma_{i,g,b}^{\mathrm{c}} + w_k\gamma_{i+1,g,b}^{\mathrm{c}}, \qquad w_k = \frac{\lambda_k - \lambda_i^{\mathrm{c}}}{\lambda_{i+1}^{\mathrm{c}} - \lambda_i^{\mathrm{c}}}. \tag{38}$$

The endpoint control values are used outside the interpolation intervals. Since the calibrated values lie in $[0,1]$, the interpolated inheritance coefficients also satisfy $0 \leq \gamma_{k,g,b} \leq 1$.

For a fixed model, sampling schedule, and conversion configuration, the inheritance coefficients are precomputed for all denoising steps before inference. Selective inheritance therefore requires only coefficient lookup and residual scaling during sampling.

C.2 Residual-State Representation under Changing Thresholds

The inheritance rule in Eq. equation 11 assumes that residual potentials are represented in the same activation space across denoising steps. When the internal state is stored in a threshold-dependent representation, it is converted to this common representation before inheritance.

Let u_k denote the stored final state at denoising step k, and let $\kappa_k > 0$ denote the element-wise conversion factor to the activation representation used in the main text:

$$r_k = \kappa_k \odot u_k. \tag{39}$$

Since reverse sampling proceeds from step k to step $k-1$, selective inheritance gives

$$v_{k-1}[0] = \gamma_{k-1} r_k. \tag{40}$$

If the initial state of step $k-1$ is stored internally as $u_{k-1}[0]$, the corresponding update is

$$u_{k-1}[0] = \gamma_{k-1} \left(\kappa_k \odot \kappa_{k-1}^{-1}\right) \odot u_k. \tag{41}$$

When the stored state already uses the same activation representation across denoising steps, $\kappa_k = \kappa_{k-1}$ and Eq. equation 41 reduces to direct residual scaling. Thus, threshold changes affect only the internal representation of the state, while the residual used by potential conservation, selective inheritance, and threshold adaptation remains defined in the common activation representation.

D Residual-Informed Threshold Adaptation

D.1 Historical Activation Statistics

We first describe how the historical statistics used for threshold adaptation are obtained from completed denoising steps. At conversion position ℓ and denoising step k, the conservation relation gives

$$a_{\ell,k} + v_{\ell,k}[0] = y_{\ell,k} + r_{\ell,k}, \tag{42}$$

and therefore the activation received at this position can be recovered as

$$\widetilde{a}_{\ell,k} = y_{\ell,k} + r_{\ell,k} - v_{\ell,k}[0]. \tag{43}$$

When the conservation relation holds exactly, $\widetilde{a}_{\ell,k} = a_{\ell,k}$. Importantly, $\widetilde{a}_{\ell,k}$ is an observation along the actual sampling trajectory rather than an activation from the reference model; discrepancies propagated from preceding denoising steps therefore remain in the recovered observation.

For positions with direct affine diffusion-time modulation,

$$a_{\ell,k} = s_{\ell,k} \odot z_{\ell,k} + d_{\ell,k}, \tag{44}$$

where $s_{\ell,k}$ and $d_{\ell,k}$ are known from the diffusion-time conditioning. When $s_{\ell,k}$ can be stably inverted, we remove this known modulation before extracting the historical statistics:

$$\widetilde{z}_{\ell,k} = s_{\ell,k}^{-1} \odot \left(\widetilde{a}_{\ell,k} - d_{\ell,k}\right). \tag{45}$$

This operation removes only the explicit affine diffusion-time modulation at the current position. For positions where such inversion is not applicable, statistics are extracted directly from $\widetilde{a}_{\ell,k}$.

We use channel-wise lower and upper quantiles to characterize the activation range. Since the same prediction rule is applied to both, the following analysis uses h_k to denote either statistic observed at log-SNR λ_k.

D.2 Error Bound for Historical Extrapolation

Reverse sampling proceeds from $k = N$ to 1. Consider three consecutive denoising steps in execution order, $k+1$, k, and $k-1$, with $\lambda_{k+1} < \lambda_k < \lambda_{k-1}$. For compactness, define

$$\Delta_- = \lambda_k - \lambda_{k+1}, \qquad \Delta_+ = \lambda_{k-1} - \lambda_k, \qquad q = \frac{\Delta_+}{\Delta_-}. \tag{46}$$

The full two-point linear extrapolation from the two most recent observations is

$$\widehat{h}_{k-1} = (1+q)h_k - q h_{k+1}. \tag{47}$$

Let $h(\lambda)$ be a locally smooth function describing the statistic trend over $[\lambda_{k+1}, \lambda_{k-1}]$, and assume $|h''(\lambda)| \le M$. Suppose the historical observations satisfy

$$|h_j - h(\lambda_j)| \le \epsilon_j, \qquad j \in \{k, k+1\}. \tag{48}$$

Then the prediction error of Eq. equation 47 is bounded by

$$\left|\widehat{h}_{k-1} - h(\lambda_{k-1})\right| \le \frac{M}{2}\Delta_+(\Delta_- + \Delta_+) + (1+q)\epsilon_k + q\epsilon_{k+1}. \tag{49}$$

To see this, consider the line passing through the exact values $h(\lambda_{k+1})$ and $h(\lambda_k)$. The standard linear-interpolation remainder contributes at most $\frac{M}{2}\Delta_+(\Delta_- + \Delta_+)$ at λ_{k-1}, while the two observation errors contribute at most $(1+q)\epsilon_k + q\epsilon_{k+1}$.

The method uses a damped form of this extrapolation,

$$\widehat{h}^{\rho}_{k-1} = h_k + \rho q(h_k - h_{k+1}), \qquad 0 \le \rho \le 1, \tag{50}$$

where $\rho = 0$ reuses the latest statistic and $\rho = 1$ recovers full linear extrapolation. If the local trend additionally satisfies $|h'(\lambda)| \le L$, then

$$\left|\widehat{h}^{\rho}_{k-1} - h(\lambda_{k-1})\right| \le (1-\rho)L\Delta_+ + \rho\frac{M}{2}\Delta_+(\Delta_- + \Delta_+) + (1+\rho q)\epsilon_k + \rho q\epsilon_{k+1}. \tag{51}$$

This follows by writing Eq. equation 50 as a convex combination of reusing h_k and the full extrapolation in Eq. equation 47. A smaller ρ reduces trend extrapolation and the amplification of historical errors, while following changes in the activation range more slowly. The bound is local to the considered trajectory and does not assume identical smoothness or observation errors across stochastic trajectories.

D.3 Activation-Range and Threshold Recovery

Applying Eq. equation 50 to the lower and upper historical statistics gives the predicted values $\widehat{h}^{-}_{k-1}$ and $\widehat{h}^{+}_{k-1}$ for the next denoising step. When the statistics are maintained before the affine diffusion-time modulation in Eq. equation 44, we restore the known modulation of step $k-1$ as

$$b^{-}_{k-1} = s_{\ell,k-1} \odot \widehat{h}^{-}_{k-1} + d_{\ell,k-1}, \qquad b^{+}_{k-1} = s_{\ell,k-1} \odot \widehat{h}^{+}_{k-1} + d_{\ell,k-1}. \tag{52}$$

The corresponding activation bounds are

$$L_{k-1} = \min\left(b^{-}_{k-1}, b^{+}_{k-1}\right), \qquad U_{k-1} = \max\left(b^{-}_{k-1}, b^{+}_{k-1}\right), \tag{53}$$

where the ordering is restored after modulation because the sign of $s_{\ell,k-1}$ may differ across channels. For positions whose statistics are maintained directly in the activation representation, $\widehat{h}^{-}_{k-1}$ and $\widehat{h}^{+}_{k-1}$ are used directly as the two bounds.

We then define the predicted symmetric activation range as

$$\widehat{R}_{k-1} = \max(|L_{k-1}|, |U_{k-1}|). \tag{54}$$

A fixed numerical constraint is applied before threshold recovery,

$$R_{k-1} = \mathrm{clip}\Big(\widehat{R}_{k-1}, R_{\min}, R_{\max}\Big), \tag{55}$$

to avoid degenerate or excessively extrapolated ranges.

Let $N_{\max}$ denote the maximum decoded magnitude per unit threshold under the fixed neuronal-timestep budget and decoding rule used in the preliminaries. The threshold for denoising step $k-1$ is

$$\mu_{k-1} = \frac{R_{k-1}}{N_{\max}}. \tag{56}$$

Thus, threshold adaptation changes the representable activation range while leaving the spike-generation and decoding rules unchanged.

D.4 Initialization and Causal Execution

The predictor is initialized according to the available history. Before any valid observation is available, we use an offline-calibrated initial range. With one valid observation, the latest statistic is

reused for the next denoising step, together with its known diffusion-time modulation where applicable. Once two valid observations are available, Eq. equation 50 is used. If the two historical log-SNR coordinates are identical or too close to form a stable ratio in Eq. equation 46, we fall back to the latest valid statistic.

At the beginning of denoising step k, the inheritance coefficient γ_k and threshold μ_k have already been determined from completed denoising steps and the known diffusion-time conditioning. After step k is completed, Eq. equation 43 provides the new activation observation, the historical statistics are updated, and the activation range and threshold for step $k-1$ are predicted.

The procedure is therefore causal: prediction for step $k-1$ uses only observations from completed denoising steps and diffusion-time information known in advance. In addition, the historical statistics are always updated from the recovered observation of the completed step rather than from the previous prediction. Prediction errors are therefore not directly propagated through the historical-statistic cache.

E Experimental Details

Table 4 summarizes the model and sampling configurations used in our experiments. We train the standard class-conditional DDPM and Improved-DDPM models on CIFAR-10, while using the publicly available pretrained DiT-XL/2-256 checkpoint for ImageNet. We evaluate $T \in \{4, 8\}$ neuronal timesteps for each denoising step, using the neuronal dynamics and thresholding scheme defined earlier in the paper.

Table 4: Model and sampling configurations used in the experiments.

Setting	DDPM	Improved-DDPM	DiT-XL/2-256
Dataset	CIFAR-10	CIFAR-10	ImageNet
Resolution	32×32	32×32	256×256
ANN weights	Trained by us	Trained by us	Public pretrained
Denoising steps	1,000	1,000	25
Sampler	Ancestral p-sampling	Ancestral p-sampling	Respaced ancestral p-sampling
CFG scale	1.0	–	1.5
Class-label protocol	$y_i = i \bmod 10$	$y_i = i \bmod 10$	ImageNet validation order
Neuronal timesteps T	4 / 8	4 / 8	4 / 8

F Additional Mechanistic Analyses

We further analyze three properties related to the mechanisms in Sections 4 and 5: feature variation across neighboring denoising steps, residual-based recovery of historical accumulated inputs, and temporal continuity of activation distributions.

F.1 Feature Variation across Adjacent Denoising Steps

To examine the cross-step feature variation associated with changes in the denoising operating point discussed in Sec. 4.1, we compare ANN features at adjacent denoising steps along the sampling trajectory. For sample b and network location ℓ, let

$$A = F^{\mathrm{ANN}}_{\ell,b,k+1}, \qquad B = F^{\mathrm{ANN}}_{\ell,b,k}, \tag{57}$$

where $A, B \in \mathbb{R}^{C\times H\times W}$ denote the features at two adjacent denoising steps. We use the shared normalization

$$n_{b,\ell,k} = \mathrm{mean}(|A|) + \mathrm{mean}(|B|) + \varepsilon, \qquad \varepsilon = 10^{-8}. \tag{58}$$

We first measure the position-wise variation while preserving the channel and spatial correspondence between the two features:

$$d_{\mathrm{pos}}(b,\ell,k) = \frac{\mathrm{mean}_{c,h,w}\,|A_{c,h,w} - B_{c,h,w}|}{n_{b,\ell,k}}. \tag{59}$$

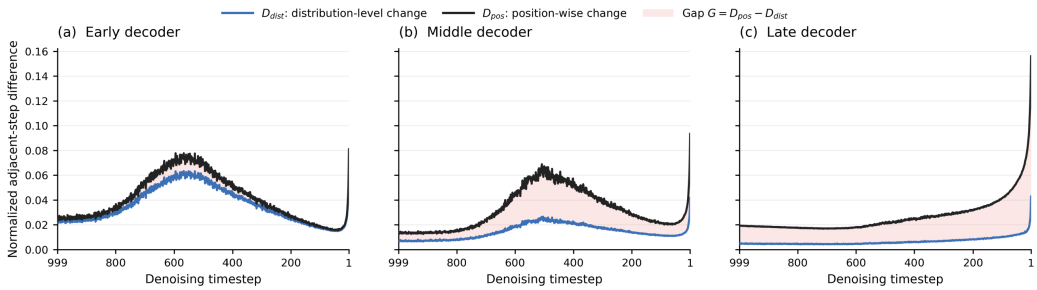


Figure 5: Distribution-level and position-wise feature variation between adjacent ANN denoising steps at representative decoder locations. The shaded gap $D_{\text{gap}} = D_{\text{pos}} - D_{\text{dist}}$ measures the additional position-wise variation beyond the channel-wise distribution change.

Averaging over $N = 32$ ANN sampling trajectories gives

$$D_{\text{pos}}(\ell, k) = \frac{1}{N} \sum_{b=1}^{N} d_{\text{pos}}(b, \ell, k). \tag{60}$$

This measure preserves the position-wise correspondence of the feature maps and is therefore sensitive to changes or rearrangements across spatial positions.

We separately measure changes in the channel-wise feature distributions. For each channel, the $H \times W$ spatial values are flattened and sorted as

$$\widetilde{A}^{(c)} = \text{sort}\left(A^{(c)}_{\text{flat}}\right), \qquad \widetilde{B}^{(c)} = \text{sort}\left(B^{(c)}_{\text{flat}}\right). \tag{61}$$

We then define

$$d_{\text{dist}}(b, \ell, k) = \frac{\text{mean}_{c,q} \left| \widetilde{A}^{(c)}_q - \widetilde{B}^{(c)}_q \right|}{n_{b,\ell,k}}, \tag{62}$$

and

$$D_{\text{dist}}(\ell, k) = \frac{1}{N} \sum_{b=1}^{N} d_{\text{dist}}(b, \ell, k). \tag{63}$$

For each channel, the numerator corresponds to the one-dimensional Wasserstein-1 distance between the two empirical spatial distributions, which is then averaged across channels. No sorting is performed across channels or samples.

We further define

$$D_{\text{gap}}(\ell, k) = D_{\text{pos}}(\ell, k) - D_{\text{dist}}(\ell, k), \tag{64}$$

which measures the additional position-wise variation not explained by the channel-wise distribution change. Since sorting gives the minimum absolute-difference matching in one dimension,

$$0 \le D_{\text{dist}}(\ell, k) \le D_{\text{pos}}(\ell, k) \le 1. \tag{65}$$

The ordering itself therefore follows from the metric definition; the quantities of interest are the magnitude and temporal structure of D_{gap}, and how they vary across network locations.

Figure 5 shows that the gap between distribution-level and position-wise variation remains non-negligible along the denoising trajectory, while its magnitude and temporal pattern vary across network locations. Thus, adjacent denoising steps can retain similar channel-wise feature distributions while exhibiting larger changes at corresponding spatial positions. This observation is consistent with the changes in the denoising operating point considered in Sec. 4.1: similarity at the distribution level does not imply position-wise feature alignment, so full inheritance can become less reliable as the denoising trajectory evolves.

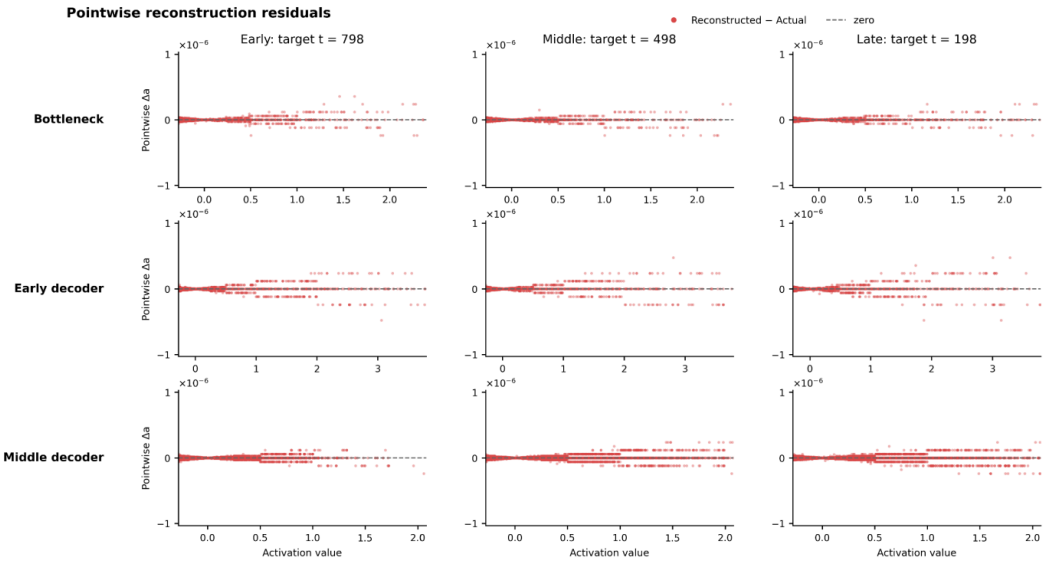


Figure 6: Pointwise verification of accumulated-input recovery at representative network locations and denoising stages. Each point shows the actual accumulated SNN input $a_{\ell,k,j}$ and its recovery error $\Delta a_{\ell,k,j} = \widetilde{a}_{\ell,k,j} - a_{\ell,k,j}$. The dashed line denotes exact recovery.

F.2 Numerical Verification of Historical Input Observation

Sec. 5.1 recovers the accumulated SNN input of a completed denoising step from quantities available after the neuronal timesteps. For network location ℓ and denoising step k, let $a_{\ell,k}$ denote the actual accumulated SNN input, $y_{\ell,k}$ the spike-decoded output, $v_{\ell,k}[0]$ the initial potential, and $r_{\ell,k}$ the final residual potential. From potential conservation,

$$v_{\ell,k}[0] + a_{\ell,k} = y_{\ell,k} + r_{\ell,k}, \tag{66}$$

the accumulated input can be recovered as

$$\widetilde{a}_{\ell,k} = y_{\ell,k} + r_{\ell,k} - v_{\ell,k}[0]. \tag{67}$$

We numerically verify this relation by directly comparing $\widetilde{a}_{\ell,k}$ with the recorded $a_{\ell,k}$ at the bottleneck, early decoder, and middle decoder. For each tensor element j, we define the pointwise recovery error as

$$\Delta a_{\ell,k,j} = \widetilde{a}_{\ell,k,j} - a_{\ell,k,j}. \tag{68}$$

As shown in Figure 6, the recovery errors remain close to zero over the examined activation range. Among the nine displayed cases, covering three network locations and three denoising stages, the maximum absolute error is 9.54×10^{-7}. Over the complete 1,000-step probe at the same three locations, the maximum absolute recovery error is 2.38×10^{-6}. These results verify that the accumulated SNN input of a completed denoising step can be recovered from quantities already available after the neuronal timesteps up to numerical precision, providing the historical range observations used in Sec. 5.

F.3 Temporal Evolution of Accumulated-Input Distributions

Sec. 5.2 predicts the current activation range from the two most recent historical range observations. We therefore examine how the actual accumulated SNN inputs evolve across adjacent denoising steps. For network location ℓ and denoising step k, let

$$a_{\ell,k} \in \mathbb{R}^{B\times C\times H\times W} \tag{69}$$

denote the accumulated SNN input of the completed denoising step. We analyze the bottleneck, early decoder, and middle decoder over three representative step triplets, $(800, 799, 798)$, $(500, 499, 498)$,

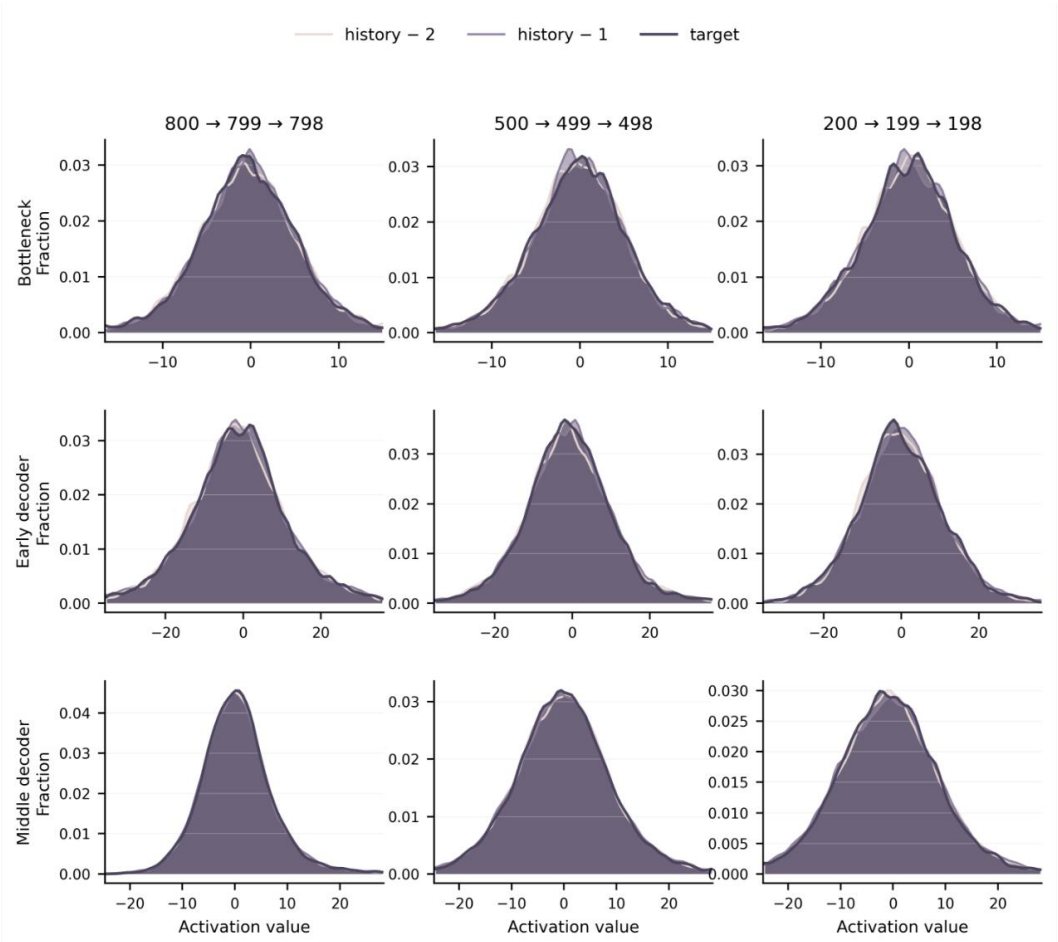


Figure 7: Empirical marginal distributions of the actual accumulated SNN inputs across three adjacent denoising steps. History-2, history-1, and target denote the two preceding steps and the latest actual observation in each three-step triplet, respectively. All three curves at the same network location use shared histogram bins.

and $(200, 199, 198)$, using $T = 8$, batch size $B = 1$, and seed 42. Here, the latest step in each triplet is referred to as the target only for notational convenience; it is an actual observation rather than a predicted distribution.

For each network location, we flatten the accumulated input as

$$\mathbf{a}_{\ell,k} = \mathrm{vec}(a_{\ell,k}) \in \mathbb{R}^{N_\ell}, \tag{70}$$

and construct an empirical histogram using 80 equal-width bins shared by all displayed denoising steps at that location. The probability mass in bin m is

$$p_{\ell,k,m} = \frac{1}{N_\ell} \sum_{j=1}^{N_\ell} \mathbb{I}\left[\tau_{\ell,m} \le [\mathbf{a}_{\ell,k}]_j < \tau_{\ell,m+1}\right]. \tag{71}$$

The histogram is smoothed for visualization with a Gaussian kernel of $\sigma = 1.2$ bins and then renormalized to unit mass, yielding $\bar{p}_{\ell,k,m}$. Thus, the vertical axis in Figure 7 represents probability mass per bin rather than probability density.

As shown in Figure 7, the empirical distributions of adjacent denoising steps remain strongly overlapping across the examined network locations and denoising stages, while their shapes change gradually along the trajectory. This indicates short-range continuity at the distribution level. It does not imply that corresponding tensor elements remain unchanged across steps, since spatial and channel-wise values may vary while the marginal distributions remain similar.

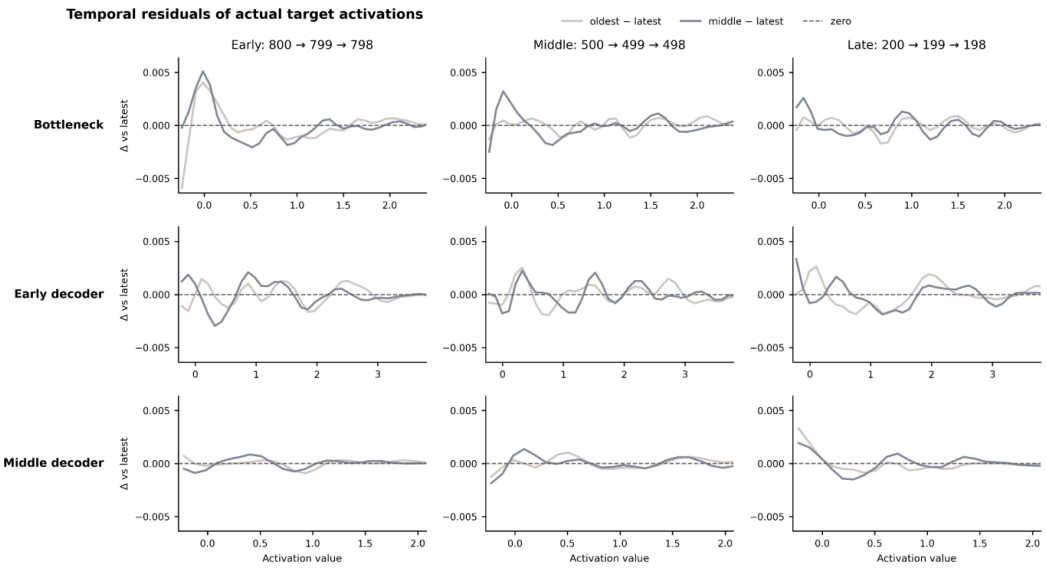


Figure 8: Temporal differences in the accumulated-input distributions shown in Figure 7. Each curve shows the difference between the smoothed empirical distribution of a preceding step and that of the latest step in the same three-step triplet. The dashed line denotes zero change in probability mass.

To make the smaller cross-step changes more visible, we additionally use the latest observation in each triplet as a reference and subtract its smoothed empirical distribution from those of the two preceding steps. Let w index the three step triplets. We define

$$\Delta p^{\mathrm{old}}_{\ell,w,m} = \bar{p}_{\ell,k_{\mathrm{old}},m} - \bar{p}_{\ell,k_{\mathrm{latest}},m}, \tag{72}$$

$$\Delta p^{\mathrm{mid}}_{\ell,w,m} = \bar{p}_{\ell,k_{\mathrm{mid}},m} - \bar{p}_{\ell,k_{\mathrm{latest}},m}. \tag{73}$$

These differences measure changes in empirical probability mass across activation values and do not preserve pointwise correspondence between tensor elements.

Figure 8 exposes the cross-step changes that are less visible in the strongly overlapping distributions of Figure 7. The differences remain structured and vary with both network location and denoising stage, showing that the accumulated-input distributions evolve gradually rather than remaining unchanged. Together, the two analyses indicate that recent historical observations retain useful short-range distributional continuity while also reflecting the ongoing range evolution. This behavior is consistent with using the two most recent historical range observations to estimate the current range in Sec. 5.2. The analysis characterizes the temporal structure used by the predictor and does not by itself establish the accuracy of the specific extrapolation rule.