SPIKEGAN: SPIKE-BASED GENERATIVE ADVERSARIAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks (SNNs) offer a route to energy-efficient generative modeling through event-driven computation, but adversarial learning must account for discrete spike representations and temporal output variability. We introduce SpikeGAN, which jointly trains an auxiliary artificial neural network (ANN) generator and an SNN generator with a shared critic. Continuous ANN samples shape the critic's representation and provide distribution-level guidance to the SNN. A temporal objective supplies direct adversarial feedback to each SNN output step, so intermediate generated samples receive their own training signal. We derive a sufficient covariance condition under which per-step supervision has lower gradient variance than average-output supervision. Experiments on CIFAR-10, CelebA, and LSUN bedroom evaluate generation quality and efficiency, while controlled ablations and multi-seed diagnostics assess the training components. On CIFAR-10, SpikeGAN achieves an FID of 15.19 with four time steps and 4.30M generator parameters, compared with the reported SDDPM result of 16.89 with 63.61M parameters, a parameter-count ratio of approximately $14.8\times$. These results support joint distribution guidance and temporal supervision for compact spike-based generative modeling.

1 INTRODUCTION

Spiking neural networks (SNNs) communicate through sparse binary events and naturally model temporal dynamics (Maass, 1997; Gerstner & Kistler, 2002; Izhikevich, 2003). Their event-driven computation is well matched to neuromorphic processors, including Loihi (Davies et al., 2018), TrueNorth (Akopyan et al., 2015), Tianjic (Pei et al., 2019), and Darwin (Ma et al., 2017). Although SNNs have shown promising performance in classification (Yao et al., 2024; Wang et al., 2025b; Yao et al., 2025), speech recognition (Wu et al., 2020; Wang et al., 2025a), and object detection (Su et al., 2023), their generative capabilities remain comparatively underexplored, largely due to the difficulty of modeling complex data distributions with discrete spikes.

Unlike classification, generation requires a model to approximate the structure and diversity of a high-dimensional data distribution. Existing SNN generators adapt variational autoencoders, diffusion models, or generative adversarial networks (GANs) (Kamata et al., 2022; Kotariya & Ganguly, 2022; Cao et al., 2024; Feng et al., 2024). Diffusion-based methods can improve sample quality but require repeated denoising, whereas GANs generate samples in a single forward pass (Goodfellow et al., 2014; Gulrajani et al., 2017; Zhang et al., 2025). GANs are therefore attractive for efficient SNN generation, although their adversarial training is sensitive to the discrete and temporal properties of spikes.

Two challenges are particularly important. First, an ANN generator uses continuous activations, whereas an SNN generator conveys intermediate information through binary firing events. Critic feedback must therefore pass through non-differentiable spikes using surrogate gradients, which can make direct alignment between SNN-generated and real distributions difficult. Second, an SNN produces a sequence of outputs before temporal averaging. Supervising only the final average may hide poor or inconsistent intermediate samples caused by reset-and-fire dynamics. These challenges motivate continuous distribution guidance and per-step SNN supervision.

ANN guidance motivates distillation, but prior SNN methods transfer discriminative logits or features (Kushawaha et al., 2021; Takuya et al., 2021; Xu et al., 2023; Guo et al., 2023).

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.

Generative learning instead requires distribution-level guidance because multiple outputs may be valid samples. Motivated by this distinction, we propose SpikeGAN, which jointly trains an auxiliary ANN generator and an SNN generator with a shared ANN critic. Continuous ANN samples contribute to the adversarial representation without imposing point-wise imitation, while a temporal objective supervises every SNN output step.

We provide conditional analyses of shared-critic transport gradients and the variance of per-step supervision, together with multi-seed diagnostics. Experiments on CIFAR-10, CelebA, and LSUN bedroom evaluate generation quality and efficiency. On CIFAR-10, SpikeGAN obtains an FID of 15.19 with 4.30M generator parameters and four time steps, compared with the reported SDDPM result of 16.89 with 63.61M parameters and eight steps.

The main contributions are summarized as follows:

- We identify the representation gap and temporal inconsistency as two key obstacles to adversarial SNN generation. To bridge the representation gap, we jointly train ANN and SNN generators with a shared critic, allowing continuous ANN samples to provide distribution-level guidance without pointwise imitation.
- We introduce per-step adversarial supervision to directly regularize the SNN output at every simulation step. We further derive a sufficient covariance condition under which per-step supervision yields lower gradient variance than average-output supervision, providing a theoretical rationale for its stabilizing effect.
- We evaluate SpikeGAN on CIFAR-10, CelebA, and LSUN bedroom using DCGAN and BigGAN backbones. SpikeGAN achieves an FID of 15.19 on CIFAR-10 with 4.30M generator parameters and four simulation steps. Ablation studies assess the contributions of joint training and temporal supervision, while multi-seed diagnostics support the gradient-variance reduction associated with per-step supervision.

2 Related Work

2.1 Spiking Generative Models

Existing SNN generative models broadly follow three families: VAE-based, diffusion-based, and GAN-based approaches. FSVAE (Kamata et al., 2022) constructs a latent space with an autoregressive SNN. Connecting autoencoding and diffusion, Spiking-Diffusion (Liu et al., 2024) uses a spike-based vector-quantized autoencoder (VQ-SVAE) to learn a discrete latent space for diffusion modeling. SDDPM (Cao et al., 2024) instead uses a spiking U-Net for noise prediction. Among GAN-based models, Spiking GAN (Kotariya & Ganguly, 2022) uses a fully spiking architecture with time-to-first-spike coding, while SGAD (Feng et al., 2024) applies attention-scoring decoding.

These families differ in learning and sampling mechanisms. VAE-based models learn latent representations through encoding and reconstruction, whereas diffusion-based models generate samples through iterative denoising. GAN-based models learn through adversarial feedback and require only a generator forward pass at inference, but their training is sensitive to the mismatch between spike-based states and continuous images. Within this line of spiking generative models, SpikeGAN combines joint ANN–SNN training through a shared critic with per-step adversarial supervision, addressing distribution-level guidance and temporal output consistency in a unified framework.

2.2 Generative Adversarial Networks

GANs (Goodfellow et al., 2014) learn generative distributions through competition between a generator and a discriminator. Subsequent work examined how the adversarial objective shapes optimization, including gradient behavior when real and generated distributions have little overlap (Arjovsky & Bottou, 2017). Wasserstein GAN (WGAN) (Arjovsky et al., 2017) introduced a Wasserstein objective to provide informative distribution-level feedback. WGAN-GP (Gulrajani et al., 2017) developed gradient-penalty regularization for the critic, providing a practical basis for Wasserstein adversarial training.

Architectural developments include DCGAN (Radford et al., 2015), which established convolutional GAN design principles, and BigGAN (Brock et al., 2018), which studied large-scale class-

conditional synthesis. SpikeGAN adopts the Wasserstein adversarial objective with gradient penalty for both its DCGAN and BigGAN variants. This shared formulation supports joint ANN–SNN training and temporal supervision, while the Wasserstein formulation in Sec. 3.2 provides the basis for our conditional analysis.

2.3 Knowledge Distillation in SNNs

Knowledge distillation transfers teacher knowledge to a student, with soft output targets providing information beyond hard class labels (Hinton et al., 2015). In SNNs, ANN-guided distillation has been used to improve learning and support low-latency inference (Kushawaha et al., 2021; Takuya et al., 2021). Xu et al. (2023) further study distillation for constructing deep SNNs from ANNs. These studies establish teacher supervision as a useful training signal for spike-based models.

Feature-level methods transfer intermediate representations. Zhang et al. (2024) align ANN and SNN features through similarity-sensitive contrastive learning based on mutual information maximization. Yang et al. (2025) align rate-based features through hybrid block-wise replacement, using intermediate ANN–SNN models to bridge their feature spaces. Together, these methods show how ANN representations can guide SNN optimization at different depths.

Joint training provides another form of ANN–SNN knowledge transfer. Joint A-SNN (Guo et al., 2023) combines multi-branch self-distillation with weight factorization, sharing singular vectors while learning branch-specific singular values. SpikeGAN uses joint training for generative distribution learning: ANN and SNN samples train a shared adversarial critic, whose feedback guides both generators. The connection between branches operates through sample distributions and critic feedback, whereas the methods above emphasize output or feature supervision and parameter coupling. Per-step adversarial supervision guides the SNN's temporal generation process.

3 Preliminaries

3.1 Spiking Neuron Model

We use the discrete-time leaky integrate-and-fire (LIF) neuron

$$\begin{aligned} u_t^l &= \tau\big(u_{t-1}^l - V_{th} s_{t-1}^l\big) + W^l s_t^{l-1}, \\ s_t^l &= H\big(u_t^l - V_{th}\big), \end{aligned} \tag{1}$$

where $u_t^l \in \mathbb{R}^{n_l}$ and $s_t^l \in \{0,1\}^{n_l}$ are the membrane-potential and spike-output vectors for the n_l neurons in layer l at time t; W^l is the synaptic weight matrix; $\tau \in (0,1]$ is the decay factor; and V_{th} is the firing threshold. The subtraction of $V_{th} s_{t-1}^l$ implements a soft reset after a spike. The element-wise Heaviside function $H(\cdot)$ has zero derivative away from the threshold and is non-differentiable at it, so we use a triangular surrogate derivative (Deng et al., 2022) during backpropagation.

3.2 GANs and Wasserstein Distance

GANs train a generator G against a discriminator D through the minimax objective:

$$\min_G \max_D \mathbb{E}_{x\sim P_{\mathrm{r}}}[\log D(x)] + \mathbb{E}_{\tilde{x}\sim P_{\mathrm{g}}}[\log(1 - D(\tilde{x}))], \tag{2}$$

where P_{r} and P_{g} are the real and generated distributions, and $D(x) \in (0,1)$ estimates the probability that x is real.

The Wasserstein-1 distance measures discrepancies between real and generated distributions. Its dual formulation replaces the probabilistic discriminator with a real-valued critic and provides our adversarial objective (Arjovsky et al., 2017). For an SNN generator G_s, let P_s denote its output distribution; Sec. 4.1 distinguishes per-step and temporally averaged outputs. The objective is

$$\begin{aligned} &\min_{G_s} W(P_{\mathrm{r}}, P_s), \\ W(P_{\mathrm{r}}, P_s) &= \sup_{\|f\|_L \le 1} \mathbb{E}_{x\sim P_{\mathrm{r}}}[f(x)] - \mathbb{E}_{x\sim P_s}[f(x)]. \end{aligned} \tag{3}$$

Here, $\|f\|_L \leq 1$ restricts f to 1-Lipschitz functions. A neural critic, denoted by D below, approximates the supremum in Eq. (3). WGAN uses weight clipping to constrain the critic (Arjovsky et al., 2017), whereas WGAN-GP regularizes its input-gradient norm toward one on interpolated samples (Gulrajani et al., 2017). Both our DCGAN and BigGAN variants use the Wasserstein objective with gradient penalty.

4 METHOD

We first analyze why direct adversarial learning is difficult for an SNN generator. We then formulate joint ANN–SNN training with a shared critic and introduce per-step temporal supervision. Finally, we give conditional analyses of critic sharing and gradient variance.

4.1 PROBLEM ANALYSIS

Let P_r denote the real-data distribution and P_a the distribution induced by the ANN generator. For latent input $z \sim P_z$, let $x_s^t = G_s^t(z)$ be the SNN output at step t and let P_s^t denote its distribution. All temporal outputs share the same latent input z. The inference output is $\bar{x}_s = T^{-1}\sum_{t=1}^{T} x_s^t$, whose induced distribution is denoted by P_s^{avg}. Thus, P_s^{avg} is the distribution of an averaged output, not the arithmetic mean of the distributions P_s^t.

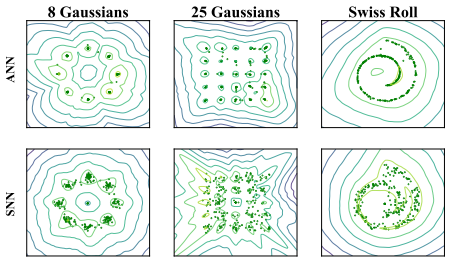


Figure 1: Toy WGAN critic surfaces and samples from ANN (top) and SNN (bottom) generators.

To visualize the optimization difficulty, we train ANN and SNN-based WGANs on toy distributions and plot the corresponding critic surfaces in Fig. 1. The ANN generator learns samples concentrated around the target distribution, whereas the SNN generator produces a more scattered distribution. This phenomenon indicates that the SNN generator is not merely a lower-capacity version of the ANN generator; it faces a different optimization geometry caused by spike discretization and temporal dynamics.

We attribute this gap to two factors. First, SNNs transmit information through binary spikes, while natural images are generated from continuous manifolds. The resulting discretization makes the SNN-generated distribution less smooth than the ANN-generated distribution and enlarges the direct Wasserstein distance to P_r. Second, supervising the temporally averaged output constrains P_s^{avg} but does not directly align each per-step distribution P_s^t with P_r. A plausible average can therefore coexist with poor intermediate outputs. These observations motivate a framework that uses ANN distribution guidance to reduce the SNN-real gap and temporal constraints to regularize each SNN output step.

4.2 SPIKEGAN

SpikeGAN jointly trains an ANN generator G_a and an SNN generator G_s with a shared critic. The distribution-level target for the average-output variant is

$$\min_{G_a, G_s} W(P_r, P_a) + W(P_r, P_s^{\text{avg}}). \tag{4}$$

Equation (4) expresses the goal of matching both generated distributions to real data. In training, the branches are coupled through a shared critic: both contribute generated samples to critic updates and receive its adversarial feedback. The ANN branch is optimized jointly with the SNN branch, providing a continuously valued source of samples that shapes the critic's learned representation. This interaction provides distribution-level guidance to the SNN through critic feedback, while each generator retains its own parameters and the SNN retains its temporal dynamics. The shared-critic training criterion is specified below.

Temporal constraint. Optimizing only P_s^{avg} allows errors at different time steps to cancel in the temporal average, so the final output may appear plausible even when individual states drift. We

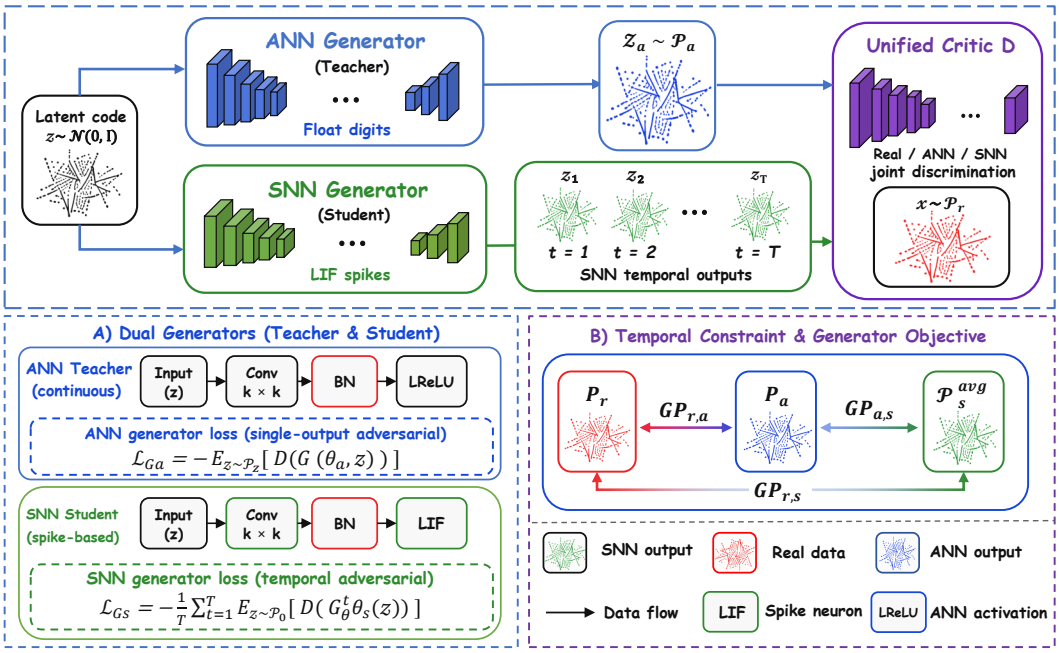


Figure 2: Overview of SpikeGAN. Given latent input z, the ANN branch produces $x_a = G_{\theta_a}(z)$ and the SNN branch produces $x_s^t = G_{\theta_s}^t(z)$ at each step. A shared critic scores real and generated samples; the SNN loss averages per-step adversarial losses. GP paths in (B) denote the gradient penalties in Eq. (7). Only the SNN generator is retained for inference.

therefore supervise every temporal distribution and replace the average-output term with

$$\min_{G_a, G_s} W(P_r, P_a) + \frac{1}{T}\sum_{t=1}^{T} W(P_r, P_s^t). \tag{5}$$

Equation (5) targets alignment of every temporal marginal P_s^t with real data. Each Wasserstein term admits its own optimal critic. For clarity, the critic and generator losses below use equal branch weights; Appendix B.2 specifies the DCGAN training weights. A shared critic D maximizes

$$\mathcal{J}_D = 2\mathbb{E}_{x\sim P_r}[D(x)] - \mathbb{E}_{x_a\sim P_a}[D(x_a)] - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{x_s^t\sim P_s^t}[D(x_s^t)]. \tag{6}$$

For the mixture $Q = \frac{1}{2}P_a + \frac{1}{2T}\sum_t P_s^t$, the ideal 1-Lipschitz supremum of $\mathcal{J}_D$ equals $2W(P_r, Q)$, bounded above by the sum in Eq. (5). A common optimal critic yields equality (Appendix A.3). Each step receives direct feedback, while the shared score measures mixture alignment. The coefficient two balances unit ANN weight and total unit SNN weight. To regularize the critic's input-gradient norm along sampled paths, we minimize

$$L_D = -\mathcal{J}_D + \lambda_{ra}\mathcal{GP}_{ra} + \lambda_{rs}\mathcal{GP}_{rs}^{\text{avg}} + \lambda_{as}\mathcal{GP}_{as}^{\text{avg}}. \tag{7}$$

Here, $\bar{x}_s = T^{-1}\sum_t x_s^t$ has distribution P_s^{avg}, $\mathcal{GP}_{ra} = \mathcal{GP}(P_r, P_a)$, $\mathcal{GP}_{rs}^{\text{avg}} = \mathcal{GP}(P_r, P_s^{\text{avg}})$, and $\mathcal{GP}_{as}^{\text{avg}} = \mathcal{GP}(P_a, P_s^{\text{avg}})$. The penalties use random line interpolations between real, ANN, and temporally averaged SNN samples; the ANN and SNN branches share the same latent input. These paths regularize the critic around the generated outputs and between the two branches. Temporal supervision remains per-step in Eq. (6) and the SNN generator loss, while the gradient penalties operate on the averaged SNN output.

For each sample pair (x, y) on a specified path, we draw $\epsilon \sim \mathcal{U}[0, 1]$ and form $\hat{x} = \epsilon x + (1-\epsilon)y$. The corresponding penalty is the expectation of $(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2$ over the sampled pairs and interpolation coefficients. The three weights λ_{ra}, λ_{rs}, and λ_{as} control the respective penalties.

Architecture and training. The architecture is shown in Fig. 2. Both generators use the same high-level convolutional upsampling design, but the SNN branch replaces continuous activations with LIF modules. The critic is an ANN and is shared by the two generated sources. Algorithm 1 summarizes the training procedure. During each critic update, real, ANN-generated, and per-step SNN-generated samples are evaluated in the same minibatch. After n_{critic} critic updates, both generators are updated once. The ANN generator and critic are training-only components; after optimization, only the SNN generator is retained. The ANN loss and every per-step SNN loss use the same critic. The two branches may use the same latent minibatch, but no point-wise equality between their outputs is imposed. This design avoids a separate critic for each time step and makes the critic representation depend on real, continuous generated, and spiking generated samples in every iteration.

Algorithm 1: SpikeGAN training.

Input: P_r, $P(z)$, T, n_{critic}, batch size m
Output: SNN generator G_{θ_s}

```
Initialize w, θ_a, and θ_s.
while not converged do
    for k = 1, ..., n_critic do
        Sample real data and latent codes.
        Generate ANN samples.
        Generate SNN samples at all T steps.
        Form the three interpolation batches.
        Compute L_D using Eqs. (6)–(7).
        Update w with Adam.
    end for
    Sample a new latent minibatch.
    Compute L_{G_a} and L_{G_s}.
    Update θ_a and θ_s with Adam.
end while
return G_{θ_s}
```

For clarity, the generator-side losses can be written as

$$\begin{aligned} L_{G_a} &= -\mathbb{E}_{z\sim P(z)}[D(G_{\theta_a}(z))], \\ L_{G_s} &= -\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{z\sim P(z)}[D(G^t_{\theta_s}(z))]. \end{aligned} \tag{8}$$

The two losses share the critic while updating separate generator parameters. The ANN term evaluates one continuous output, whereas the SNN term averages adversarial feedback across all T temporal outputs. Every SNN step thus contributes to updating the generator.

4.3 Conditional Analysis of the Training Objective

The shared critic and per-step supervision raise two questions: when can one critic provide compatible transport directions across distribution pairs, and when does averaging per-step gradients reduce their variance? Proposition 1 addresses the first through a common optimal potential. Proposition 2 addresses the second through a covariance condition on the implemented gradient estimators. Together, they explain the geometry and optimization of the training components in Sec. 4.

Proposition 1. *(Proof in Appendix A.1) Let $\mathbb{P}_s$, $\mathbb{P}_a$, and $\mathbb{P}_r$ be distributions on a compact convex subset $\mathcal{X} \subset \mathbb{R}^d$. Assume the geodesic-collinearity condition*

$$W(\mathbb{P}_s, \mathbb{P}_r) = W(\mathbb{P}_s, \mathbb{P}_a) + W(\mathbb{P}_a, \mathbb{P}_r) \tag{9}$$

holds. Assume a differentiable 1-Lipschitz $f^* : \mathbb{R}^d \to \mathbb{R}$ satisfies $\mathbb{E}_{P_j} f^* - \mathbb{E}_{P_i} f^* = W(P_i, P_j)$ for every $(i, j) \in \{(s, a), (a, r), (s, r)\}$. Let π^*_{ij} be an optimal coupling with $\pi^*_{ij}\{x_i = x_j\} = 0$. For $0 < t < 1$, define $x_t^{(ij)} = tx_j + (1 - t)x_i$. Then

$$\mathbb{P}_{(x_i,x_j)\sim\pi^*_{ij}}\left[\nabla f^*(x_t^{(ij)}) = \frac{x_j - x_i}{\|x_j - x_i\|}\right] = 1. \tag{10}$$

Proposition 1 identifies when critic directions agree across the three pairs; applying it to all P_s^t requires the same potential at every step. We next ask when per-step supervision reduces gradient variability. At fixed critic and generator parameters, let g_t and g_{avg} be the mini-batch surrogate gradients of $-D(G_s^t(z))$ and $-D(\bar{x}_s(z))$, using the same batch and surrogate rule. Proposition 2 compares their sampling variability.

Proposition 2. *(Proof in Appendix A.2) For the stochastic gradients defined above, let $T \geq 2$ and $g_{\text{temp}} = T^{-1} \sum_{t=1}^{T} g_t$. Assume finite second moments and $\text{Tr}(\text{Cov}(g_t)) = \sigma^2 > 0$ for every t, and define*

$$\bar{\rho} = \frac{2}{T(T-1)\sigma^2} \sum_{1 \leq i < j \leq T} \text{Tr}(\text{Cov}(g_i, g_j)). \tag{11}$$

If $\text{Tr}(\text{Cov}(g_{\text{avg}})) = \alpha\sigma^2$ and

$$\alpha \geq \frac{1 + (T-1)\bar{\rho}}{T}, \tag{12}$$

then

$$\text{Tr}(\text{Cov}(g_{\text{temp}})) \leq \text{Tr}(\text{Cov}(g_{\text{avg}})). \tag{13}$$

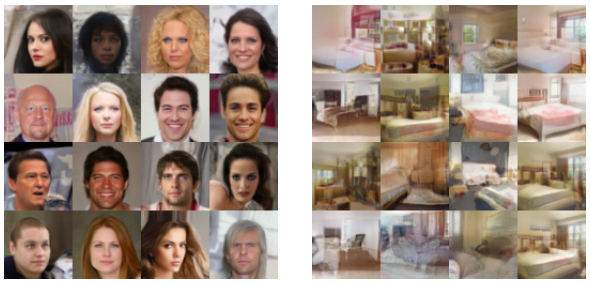
The condition permits positive cross-step covariance and guarantees no greater trace variance; a strict inequality in Eq. (12) gives strictly lower variance. The comparison fixes the model parameters.

Corollary 1 (Proposition 1). Under the conditions of Proposition 1, $\|\nabla f^*\|_2 = 1$ almost everywhere along the three coupled transport paths. Thus, a shared critic is compatible with the pairwise transport directions when the strong common-potential assumption holds. The result motivates parameter sharing.

Proposition 1 is a conditional consistency result for critic sharing, whereas Proposition 2 provides a directly testable sufficient condition for the temporal objective. Their complete proofs are given in Appendices A.1 and A.2; the assumptions and finite-run evidence are discussed separately below.

5 Experiments

5.1 Experiment Settings



(a) CelebA. (b) LSUN.

Figure 3: Generated samples: (a) CelebA; (b) LSUN bedroom.

We evaluate SpikeGAN on CIFAR-10 at 32×32 resolution (Krizhevsky et al., 2009), CelebA at 64×64 (Liu et al., 2015), and LSUN bedroom at 64×64 (Yu et al., 2015). We compare with FSVAE (Kamata et al., 2022), SGAD (Feng et al., 2024), and SDDPM (Cao et al., 2024), and report DCGAN- and BigGAN-style variants denoted by SpikeGAN† and SpikeGAN‡, respectively.

FID (Heusel et al., 2017) is the primary quality metric and is computed from 50,000 generated samples; IS (Salimans et al., 2016) is also reported for ablations. Baseline values are taken from the published studies when checkpoints are unavailable. The controlled ablation evaluates the proposed objectives under matched experimental settings.

Parameter counts and operation-based energy ratios include the inference generator only because the ANN branch and critic are discarded after training. Architectures, optimization settings, energy accounting, and diagnostic protocols are provided in Appendix B.

We estimate inference energy from measured firing rates and operation counts:

$$E_{\text{SNN}} = E_{\text{AC}} \sum_{t=1}^{T} \sum_{l} A_l^t C_l, \qquad E_{\text{ANN}} = E_{\text{MAC}} \sum_{l} C_l, \tag{14}$$

where A_l^t is the firing rate and C_l is the number of synaptic connections. The reported ratios combine measured firing activity with the operation costs in Davies et al. (2018) to estimate generator inference energy at the operation level.

Both backbones replace generator activations with LIF neurons and use $T = 4$, $\tau = 0.5$, and $V_{th} = 1$ unless stated otherwise. Both the DCGAN and BigGAN variants use the Wasserstein adversarial objective with gradient penalty and apply the shared-critic and per-step supervision strategy. Appendix B gives the layer configurations, learning rates, batch sizes, and optimizer settings.

5.2 Comparisons with the State of the Art

Table 1 compares representative SNN generators using published values. On CIFAR-10, SpikeGAN‡ obtains FID 15.19 with 4.30 million generator parameters and four steps. Relative to SDDPM, it uses $14.8\times$ fewer parameters and half the simulation steps while lowering FID by 1.70. SpikeGAN† further reduces the model to 1.07 million parameters, but its FID rises to 67.97, confirming that backbone capacity remains a major determinant of generation quality.

On CelebA, SDDPM has the lowest reported FID (25.09), followed by SpikeGAN‡ (28.88), which uses approximately half as many generator parameters. On LSUN bedroom, SpikeGAN‡ improves FID from 47.64 to 37.41 with roughly half the parameters of SDDPM. Figure 3 shows representative samples on both datasets. These results identify two practical operating points: the compact DCGAN variant prioritizes model size, while the BigGAN variant prioritizes generation quality. We treat the published baseline table as a system-level comparison and use the controlled ablation below to isolate the proposed training objectives.

Table 1: Comparison with SNN generative models. †: DCGAN backbone; ‡: BigGAN backbone.

Dataset	Model	Params (M)	Steps	FID↓
CIFAR-10	FSVAE	3.87	16	175.50
	SGAD	–	16	181.50
	SDDPM	63.61	8	16.89
	SpikeGAN†	**1.07**	4	67.97
	SpikeGAN‡	4.30	4	**15.19**
CelebA	FSVAE	6.37	16	101.60
	SGAD	–	16	151.36
	SDDPM	63.61	4	**25.09**
	SpikeGAN†	**1.13**	4	81.11
	SpikeGAN‡	32.03	4	28.88
LSUN	SDDPM	63.61	4	47.64
	SpikeGAN‡	**32.03**	4	**37.41**

5.3 Ablation Study

We conduct controlled ablations on CIFAR-10 with four time steps. Both the baseline channel setting and a variant with channels scaled by $4\times$ are included to separate objective design from capacity. Figure 4 summarizes the quality–energy trade-off. In the larger setting, joint ANN–SNN training obtains IS 4.83 and FID 60.45; adding temporal supervision improves these values to IS 4.90 and FID 55.25. The resulting 5.20-point FID reduction isolates the benefit of applying critic feedback at every time step rather than only to the averaged output. The normalized energy estimate rises by approximately 0.2–0.3 relative to the matched SNN baseline because the objective changes learned firing activity; the auxiliary ANN branch adds no inference computation.

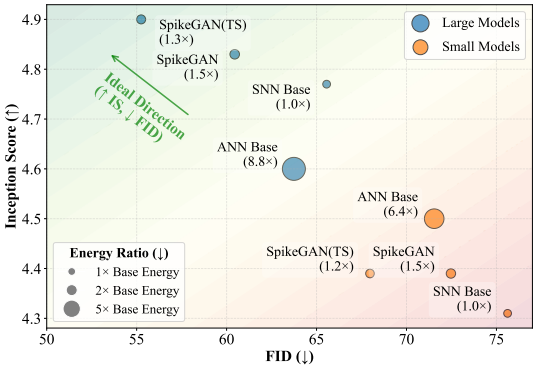


Figure 4: CIFAR-10 ablation; marker size denotes energy relative to the SNN baseline.

Under the operation-cost assumptions above, the SNN variants use approximately one seventh of the estimated compute energy of the corresponding ANN. This reduction follows from sparse accumulate operations, with the measured firing rates determining activity across layers and time steps.

Training cost. For the DCGAN architecture on CIFAR-10, memory per image increases by 1.0% (68.91 to 69.63 MB) and time per 1000 iterations by 14.6% (0.233 to 0.267 h). This overhead is confined to optimization: the ANN branch and critic are discarded afterward, leaving the deployed SNN graph, parameter count, and inference path unchanged. Training resource costs should therefore be compared separately from deployment parameter counts and inference energy.

We further examine the Wasserstein estimate and generator loss in Fig. 5. The vanilla SNN baseline converges more slowly. With joint ANN–SNN training, continuous ANN samples contribute to the critic and the Wasserstein estimate decreases more rapidly. Adding temporal supervision produces a smoother generator-loss curve, consistent with reduced step-to-step update variability.

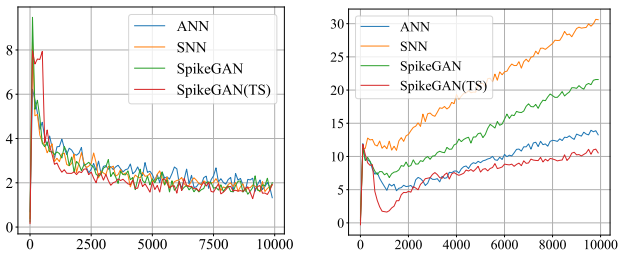


Figure 5: Training curves for the baseline and SpikeGAN: Wasserstein estimate (left) and generator loss (right).

These trajectories complement the capacity-controlled ablation.

5.4 Latent Interpolation and Temporal Stability



Figure 6: Latent-space interpolation on CelebA. The endpoints correspond to $t = 0$ and $t = 1$.

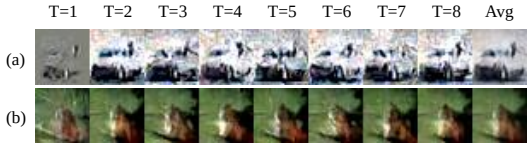


Figure 7: Per-step samples from (a) vanilla SNN-GAN and (b) SpikeGAN with temporal supervision (TS); Avg is the temporal mean.

Interpolating $z_t = (1 - t)z_1 + tz_2$ produces gradual changes in identity and appearance (Fig. 6), indicating local smoothness of the generated samples along the interpolation path.

SNN samples are formed through temporal accumulation, so unstable intermediate outputs can degrade the final average. Figure 7 shows that the vanilla SNN-GAN degrades at several steps, whereas temporal supervision produces more consistent step-wise samples and a more stable average. This supports constraining P_s^t rather than only P_s^{avg}.

5.5 Empirical Diagnostics for the Conditional Analysis

Across five CIFAR-10 runs, the temporal-gradient estimator reaches a late-window ratio of $R_{\text{var}} = 0.126 \pm 0.059$ relative to average-output supervision and remains below one at every checkpoint. Both estimators use the same critic, generator parameters, and latent mini-batches and are evaluated on the same output-layer parameter subset, isolating the difference between the two temporal objectives. The consistent reduction shows that the variance advantage is sustained throughout training rather than confined to the final iterations. Real–SNN and ANN–SNN directional agreement also increases late in training, although the latter remains modest. These observations directly support the variance mechanism in Proposition 2 and provide partial, finite-run evidence for the idealized geometry in Proposition 1. Appendix C reports the protocol, metrics, tables, and full trajectories.

6 Conclusion

SpikeGAN jointly trains ANN and SNN generators with a shared critic and applies adversarial supervision at every SNN simulation step. The BigGAN-style variant obtains an FID of 15.19 on CIFAR-10 with four steps and 4.30M inference parameters under the reported protocol. Controlled ablations demonstrate additional gains from temporal supervision, while multi-seed diagnostics show that per-step averaging yields lower output-layer gradient variance than average-output supervision. The observed geometric trends provide preliminary empirical support for the common-potential perspective and highlight its value in explaining the training dynamics of the shared critic. This framework opens promising avenues for developing higher-quality, more scalable, and energy-efficient SNN-based generative models across diverse tasks and neuromorphic hardware platforms.

AI USE STATEMENT

We used generative AI tools to assist with language editing and polishing, literature retrieval and discovery, and the development of mathematical proofs. The proof-related assistance included refining assumptions, exploring derivations, and improving the presentation of arguments. The authors take responsibility for the final content, including the accuracy of citations and the correctness of mathematical claims and proofs.

REPRODUCIBILITY STATEMENT

Section 4 specifies the model, training objectives, and theoretical assumptions, and Algorithm 1 summarizes the training procedure. Appendix A provides complete proofs and the shared-critic analysis. Appendix B describes the datasets, evaluation metrics, network configurations, optimization settings, and energy-accounting assumptions. Appendix C reports the multi-seed diagnostic protocol, gradient estimators, summary statistics, and trajectories supporting Section 5.5.

REFERENCES

Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE transactions on computer-aided design of integrated circuits and systems*, 34(10):1537–1557, 2015.

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. Pmlr, 2017.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Jiahang Cao, Ziqing Wang, Hanzhong Guo, Hao Cheng, Qiang Zhang, and Renjing Xu. Spiking denoising diffusion probabilistic models. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4900–4909. IEEE, 2024.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee micro*, 38(1):82–99, 2018.

Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. *arXiv preprint arXiv:2202.11946*, 2022.

Linghao Feng, Dongcheng Zhao, and Yi Zeng. Spiking generative adversarial network with attention scoring decoding. *Neural Networks*, 178:106423, 2024.

Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Yufei Guo, Weihang Peng, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Xuhui Huang, and Zhe Ma. Joint a-snn: Joint training of artificial and spiking neural networks via self-distillation and weight factorization. *Pattern Recognition*, 142:109639, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572, 2003.

Hiromichi Kamata, Yusuke Mukuta, and Tatsuya Harada. Fully spiking variational autoencoder. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7059–7067, 2022.

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

Vineet Kotariya and Udayan Ganguly. Spiking-gan: A spiking generative adversarial network using time-to-first-spike coding. In *2022 international joint conference on neural networks (IJCNN)*, pp. 1–7. IEEE, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ravi Kumar Kushawaha, Saurabh Kumar, Biplab Banerjee, and Rajbabu Velmurugan. Distilling spikes: Knowledge distillation in spiking neural networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4536–4543. IEEE, 2021.

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. IEEE, 2024.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

De Ma, Juncheng Shen, Zonghua Gu, Ming Zhang, Xiaolei Zhu, Xiaoqiang Xu, Qi Xu, Yangjing Shen, and Gang Pan. Darwin: A neuromorphic hardware co-processor based on spiking neural networks. *Journal of systems architecture*, 77:43–51, 2017.

Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Qiaoyi Su, Yuhong Chou, Yifan Hu, Jianing Li, Shijie Mei, Ziyang Zhang, and Guoqi Li. Deep directly-trained spiking neural networks for object detection. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6532–6542. IEEE, 2023.

Sugahara Takuya, Renyuan Zhang, and Yasuhiko Nakashima. Training low-latency spiking neural network through knowledge distillation. In *2021 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS)*, pp. 1–3. IEEE, 2021.

Shuai Wang, Dehao Zhang, Ammar Belatreche, Yichen Xiao, Hongyu Qing, Wenjie Wei, Malu Zhang, and Yang Yang. Ternary spike-based neuromorphic signal processing system. *Neural Networks*, 187:107333, 2025a.

Shuai Wang, Malu Zhang, Dehao Zhang, Ammar Belatreche, Yichen Xiao, Yu Liang, Yimeng Shan, Qian Sun, Enqi Zhang, and Yang Yang. Spiking vision transformer with saccadic attention. In *International Conference on Learning Representations*, volume 2025, pp. 24148–24169, 2025b.

Jibin Wu, Emre Yılmaz, Malu Zhang, Haizhou Li, and Kay Chen Tan. Deep spiking neural networks for large vocabulary automatic speech recognition. *Frontiers in neuroscience*, 14:199, 2020.

Qi Xu, Yaxin Li, Jiangrong Shen, Jian K Liu, Huajin Tang, and Gang Pan. Constructing deep spiking neural networks from artificial neural networks with knowledge distillation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7886–7895. IEEE, 2023.

Shu Yang, Chengting Yu, Lei Liu, Hanzhi Ma, Aili Wang, and Erping Li. Efficient ann-guided distillation: Aligning rate-based features of spiking neural networks through hybrid block-wise replacement. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10025–10035. IEEE, 2025.

Man Yao, JiaKui Hu, Tianxiang Hu, Yifan Xu, Zhaokun Zhou, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips. In *International conference on learning representations*, volume 2024, pp. 52885–52907, 2024.

Man Yao, Xuerui Qiu, Tianxiang Hu, Jiakui Hu, Yuhong Chou, Keyu Tian, Jianxing Liao, Luziwei Leng, Bo Xu, and Guoqi Li. Scaling spike-driven transformer with efficient spike firing approximation training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4): 2973–2990, 2025.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Wenchuan Zhang, Jie Zhang, Ricky Yuen-Tan Hou, Weifeng Su, and Wentao Fan. Spiking generative models based on variational autoencoder and adversarial training. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Yuhan Zhang, Xiaode Liu, Yuanpei Chen, Weihang Peng, Yufei Guo, Xuhui Huang, and Zhe Ma. Enhancing representation of spiking neural networks via similarity-sensitive contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16926–16934, 2024.

A Proofs for the Conditional Analysis

A.1 Proof of Proposition 1

Proof. Use the distributions and common potential specified in Proposition 1. The assumed relation

$$W(P_s, P_r) = W(P_s, P_a) + W(P_a, P_r) \tag{15}$$

places the ANN distribution between the SNN and real distributions in Wasserstein distance. The gradient identity follows from pairwise optimality of the common potential, as shown below.

Fix $(i, j) \in \{(s, a), (a, r), (s, r)\}$ and an optimal coupling π^*_{ij}. The orientation $\mathbb{E}_{P_j} f^* - \mathbb{E}_{P_i} f^* = W(P_i, P_j)$ agrees with higher critic scores toward real data. Define $h(x_i, x_j) = \|x_j - x_i\| - [f^*(x_j) - f^*(x_i)]$. The 1-Lipschitz property gives $h \geq 0$, while primal and dual optimality give

$$\int h \, d\pi^*_{ij} = W(P_i, P_j) - \left(\mathbb{E}_{P_j} f^* - \mathbb{E}_{P_i} f^*\right) = 0.$$

Hence $h = 0$ almost surely under π^*_{ij}. Continuity extends this equality to its support:

$$f^*(x_j) - f^*(x_i) = \|x_j - x_i\|, \quad (x_i, x_j) \in \operatorname{supp}(\pi^*_{ij}). \tag{16}$$

For a noncoincident pair in this support, set $v = x_j - x_i \neq 0$. Convexity keeps $x_t^{(ij)} = x_i + tv$ in $\mathcal{X}$ for $0 < t < 1$. The 1-Lipschitz property implies

$$f^*(x_t^{(ij)}) - f^*(x_i) \leq t\|x_j - x_i\|, \tag{17}$$

$$f^*(x_j) - f^*(x_t^{(ij)}) \leq (1 - t)\|x_j - x_i\|. \tag{18}$$

If either inequality were strict, their sum would contradict the support equality above. Hence both inequalities are tight. Differentiating the tight equality along the segment yields

$$\nabla f^*(x_t^{(ij)})^\top (x_j - x_i) = \|x_j - x_i\|. \tag{19}$$

Differentiability and the ambient 1-Lipschitz condition give $\|\nabla f^*(x_t^{(ij)})\|_2 \leq 1$. The preceding identity attains equality in Cauchy–Schwarz with positive orientation, so

$$\nabla f^*(x_t^{(ij)}) = \frac{x_j - x_i}{\|x_j - x_i\|}. \tag{20}$$

The coupling assigns zero mass to coincident pairs, so the identity holds π^*_{ij}-almost surely. Applying the argument to each of the three pairs proves the proposition and the unit-gradient corollary. For $P_s = P_s^k$, the same proof applies at step k; a shared conclusion over all steps uses a single potential satisfying the stated assumptions for every k. □

A.2 Proof of Proposition 2

Proof. Condition on fixed generator parameters θ_s and critic parameters w. For a random latent mini-batch $B = (z_1, \ldots, z_m)$, define

$$\widehat{\ell}_t(B) = -\frac{1}{m} \sum_{b=1}^{m} D_w(G^t_{\theta_s}(z_b)), \qquad \widehat{\ell}_{\text{avg}}(B) = -\frac{1}{m} \sum_{b=1}^{m} D_w\left(\frac{1}{T} \sum_t G^t_{\theta_s}(z_b)\right).$$

Let $g_t(B) = \widetilde{\nabla}_{\theta_s} \widehat{\ell}_t(B)$ and $g_{\text{avg}}(B) = \widetilde{\nabla}_{\theta_s} \widehat{\ell}_{\text{avg}}(B)$, where $\widetilde{\nabla}$ denotes backpropagation using the fixed surrogate derivative for spikes. With the same forward computation and surrogate rule for each term, linearity over the summed losses gives

$$g_{\text{temp}} = \frac{1}{T} \sum_{t=1}^{T} g_t \tag{21}$$

for the temporal loss on the same batch. All covariances below are over B, conditional on the fixed parameters. Assume finite second moments, $T \geq 2$, and equal positive per-step trace variance:

$$\operatorname{Tr}(\operatorname{Cov}(g_t)) = \sigma^2, \tag{22}$$

and define the average normalized temporal covariance

$$\bar{\rho} = \frac{2}{T(T-1)\sigma^2} \sum_{1 \le i < j \le T} \mathrm{Tr}(\mathrm{Cov}(g_i, g_j)). \tag{23}$$

Covariance bilinearity retains all cross-step terms and gives

$$\mathrm{Tr}(\mathrm{Cov}(g_{\mathrm{temp}})) = \frac{1}{T^2}\left[T\sigma^2 + 2\sum_{i<j} \mathrm{Tr}(\mathrm{Cov}(g_i, g_j))\right] = \frac{\sigma^2}{T}\big[1 + (T-1)\bar{\rho}\big]. \tag{24}$$

Let the gradient computed from the temporally averaged output satisfy

$$\mathrm{Tr}(\mathrm{Cov}(g_{\mathrm{avg}})) = \alpha\sigma^2. \tag{25}$$

Subtracting the average-output trace variance yields

$$\mathrm{Tr}(\mathrm{Cov}(g_{\mathrm{temp}})) - \mathrm{Tr}(\mathrm{Cov}(g_{\mathrm{avg}})) = \sigma^2\left[\frac{1 + (T-1)\bar{\rho}}{T} - \alpha\right].$$

Since $\sigma^2 > 0$, this difference is nonpositive precisely when

$$\alpha \ge \frac{1 + (T-1)\bar{\rho}}{T}. \tag{26}$$

This proves the claimed inequality. Equality in the condition gives equal trace variances, and strict inequality gives strictly lower temporal trace variance. Positive cross-step covariance is also permitted by this threshold. □

A.3 Shared-Critic Objective and Distribution-Level Targets

For fixed generated distributions, define the probability mixture $Q = \frac{1}{2}P_a + \frac{1}{2T}\sum_{t=1}^{T} P_s^t$. This mixture is distinct from the distribution P_s^{avg} of temporally averaged outputs. By linearity of expectation and Wasserstein duality,

$$\sup_{\|D\|_L \le 1} \mathcal{J}_D = 2 \sup_{\|D\|_L \le 1} \big(\mathbb{E}_{P_r} D - \mathbb{E}_Q D\big) = 2W(P_r, Q) \le W(P_r, P_a) + \frac{1}{T}\sum_t W(P_r, P_s^t).$$

The inequality follows by bounding each expectation difference by its separate dual supremum. If one 1-Lipschitz critic attains all the pairwise suprema, substitution gives equality. The average-output case follows by replacing the temporal mixture with P_s^{avg}. Equations (4) and (5) therefore specify pairwise distribution-level targets, while Eq. (6) uses a shared-critic criterion. With finite neural critics and the soft penalties in Eq. (7), the regularized optimization is distinct from the ideal constrained supremum.

Mixture alignment alone does not identify each component distribution. For example, let $T = 2$, $P_r = (\delta_{-1} + \delta_1)/2$, $P_a = P_r$, $P_s^1 = \delta_{-1}$, and $P_s^2 = \delta_1$. Then $Q = P_r$, so the shared supremum is zero, while the pairwise target is one under the Euclidean ground metric. Thus per-step feedback and individual distribution matching are distinct claims. Proposition 1 analyzes a conditional common-potential setting; Proposition 2 applies directly to the shared critic's stochastic gradient estimators.

B Experimental and Implementation Details

B.1 Evaluation and Energy Accounting

FID uses the standard Inception feature space. Published baseline results retain their original preprocessing and evaluation protocols.

In Eq. (14), E_{AC} and E_{MAC} are the assumed costs of an accumulate and a multiply–accumulate operation, respectively (Davies et al., 2018). Energy ratios are normalized to the SNN base model with the same backbone and include generator inference only. The operation-based estimate excludes memory access and peripheral circuitry.

B.2 Architectures and Optimization

Both backbones use Adam (Kingma & Ba, 2014) with the shared-critic objective in Eqs. (6)–(7).

For the DCGAN backbone, the SNN generator replaces ReLU activations with LIF neurons. On CIFAR-10, the base generator uses transposed convolutional blocks with channel progression $100 \to 256 \to 128 \to 64 \to C$, batch normalization, and LIF activations, giving 1.07M parameters for $C = 3$. The larger ablation variant uses hidden channels $(1024, 512, 256)$. For CelebA, an additional upsampling block is added to produce 64×64 samples. We use learning rate 1×10^{-4}, $\beta = (0.5, 0.999)$, batch size 64, five critic updates per generator update, and output clipping to $[-1, 1]$. The adversarial terms average critic scores over individual time steps, whereas the SNN endpoints of the gradient penalties use the temporally averaged image.

For the BigGAN backbone, ReLU activations in residual blocks are replaced with LIF neurons to preserve spike-driven computation. We use batch size 128 on CIFAR-10 and batch size 32 on CelebA and LSUN bedroom. The generator and discriminator learning rates are both 2×10^{-4}, with four discriminator updates per generator update. CIFAR-10 training runs for 500 epochs with exponential moving average, and normal initialization is used for both networks. All main experiments are conducted with PyTorch on NVIDIA RTX 4090 GPUs.

DCGAN loss weights. The main text presents the shared-critic objective with equal branch weights. In the DCGAN implementation, we use $w_s = 0.9$ and $w_a = 0.1$. Writing $d_r = \mathbb{E}[D(x)]$, $d_a = \mathbb{E}[D(x_a)]$, and $d_s = T^{-1} \sum_t \mathbb{E}[D(x_s^t)]$, the implemented losses are

$$\begin{aligned} L_D^{\mathrm{DCGAN}} &= w_s(d_s - d_r) + w_a(d_a - d_r) + \lambda_{ra}\mathcal{GP}_{ra} + \lambda_{rs}\mathcal{GP}_{rs}^{\mathrm{avg}} + \lambda_{as}\mathcal{GP}_{as}^{\mathrm{avg}}, \\ L_{G_s}^{\mathrm{DCGAN}} &= -w_s d_s, \qquad L_{G_a}^{\mathrm{DCGAN}} = -w_a d_a. \end{aligned} \tag{27}$$

The gradient-penalty weights remain $(10, 10, 5)$ and use the averaged SNN output. The weighted shared score corresponds to $Q_w = w_a P_a + (w_s/T) \sum_t P_s^t$: its ideal 1-Lipschitz supremum is $W(P_r, Q_w)$. Multiplying both SNN gradient estimators in Proposition 2 by w_s scales their variances by w_s^2 and preserves the variance comparison.

Table 2: Channel configuration of the SpikeGAN BigGAN generator.

Dataset	Res.	Layer	In ch.	Out ch.
CIFAR-10	32×32	Layer 1	256	256
		Layer 2	256	256
		Layer 3	256	3
CelebA/LSUN	64×64	Layer 1	1024	512
		Layer 2	512	256
		Layer 3	256	128
		Layer 4	128	3

C Detailed Diagnostics for the Conditional Analysis

The diagnostics use the CIFAR-10 training split, the DCGAN-based backbone, $T = 4$, batch size 32, 3000 training steps, three critic updates per generator update, and metric logging every 50 steps. We analyze 45 checkpoints from steps 800–3000 for seeds 7, 17, 27, 37, and 47. Each gradient-variance estimate uses six newly sampled mini-batches at a fixed checkpoint. Gradients are restricted to the parameters of the final generator block. Tables report mean$\pm$std across the five seeds; window means first average the checkpoints within each run. Figure 8 shows checkpoint-wise seed means with approximate 95% normal confidence intervals, computed as the mean ± 1.96 times the sample standard deviation divided by $\sqrt{5}$.

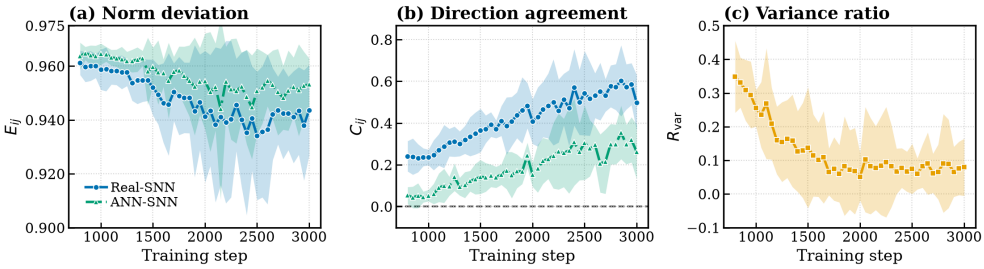


Figure 8: Diagnostic trajectories over steps 800–3000 corresponding to Tabs. 3–4. Curves show the five-seed mean and shaded 95% confidence interval.

Table 3: Common-critic diagnostics (steps 800–3000; 5 seeds, mean±std). Final: step 3000.

Path	**Fin.** $E_{ij}\downarrow$	**Mean** $E_{ij}\downarrow$	**Fin.** $C_{ij}\uparrow$	**Mean** $C_{ij}\uparrow$
Real-SNN	0.944±0.013	0.947±0.011	0.498±0.107	0.420±0.059
ANN-SNN	0.953±0.008	0.956±0.008	0.263±0.105	0.185±0.042
C_{ij} by training stage				
	Mid (800–1500)		**Late (2200–3000)**	
Real-SNN	0.285±0.036		0.535±0.091	
ANN-SNN	0.089±0.022		0.279±0.082	

Table 4: Gradient-variance and finite-critic distance diagnostics over steps 800–3000 (5 seeds, mean±std).

Metric	**Mean**	**Final**
$R_{\text{var}}\downarrow$	0.126±0.059	0.080±0.068
$\text{Tr}(\text{Cov}(g_{\text{temp}}))\downarrow$	0.075±0.021	0.047±0.044
$\text{Tr}(\text{Cov}(g_{\text{avg}}))$	1.242±0.579	0.938±0.733
Adjacent ρ_t	0.086±0.154	−0.020±0.189
$\widehat{W}(P_r, P_s)$	9.67±2.41	11.91±12.13
$\widehat{W}(P_r, P_a)$	8.03±1.53	8.65±3.63
$\widehat{W}(P_a, P_s)$	21.60±5.19	21.51±13.19

C.1 Diagnostics for Proposition 1

We compare the shared critic D with pairwise critics D_{rs}, D_{ra}, and D_{as}. For each pair (i, j), we sample

$$\hat{x}_{ij} = \epsilon x_i + (1-\epsilon)x_j, \quad \epsilon \sim U(0,1), \tag{28}$$

and monitor the finite-run Lipschitz residual

$$E_{ij} = \mathbb{E}_{\hat{x}_{ij}}\left[\left|\|\nabla_{\hat{x}_{ij}} D(\hat{x}_{ij})\|_2 - 1\right|\right] \tag{29}$$

and shared-to-pairwise gradient-direction agreement

$$C_{ij} = \mathbb{E}_{\hat{x}_{ij}}\left[\frac{\langle \nabla D(\hat{x}_{ij}), \nabla D_{ij}(\hat{x}_{ij})\rangle}{\|\nabla D(\hat{x}_{ij})\|_2 \|\nabla D_{ij}(\hat{x}_{ij})\|_2}\right]. \tag{30}$$

Table 3 and Fig. 8(a)–(b) show similar residual magnitudes. The real–SNN cosine similarity has a late-window mean of 0.420±0.059. ANN–SNN alignment increases from 0.089±0.022 over steps 800–1500 to 0.279±0.082 over steps 2200–3000, but remains modest. Thus, a shared directional structure emerges along some paths without verifying the strong common-potential assumption.

C.2 Diagnostics for Proposition 2

At each checkpoint, both gradient estimators use the same critic, generator parameters, latent mini-batches, and output-layer parameter subset. Write the fixed-critic score difference as $\widehat{W}_D(P_i, P_j) = \mathbb{E}_{P_i}[D(x)] - \mathbb{E}_{P_j}[D(x)]$. For mini-batch b, expectations are replaced by batch means and $\widetilde{\nabla}$ denotes the implemented surrogate backpropagation. We record

$$g_t^{(b)} = \widetilde{\nabla}_{\theta_{\text{out}}} \widehat{W}_D(P_r, P_s^t), \tag{31}$$

$$g_{\text{temp}}^{(b)} = \frac{1}{T} \sum_{t=1}^{T} g_t^{(b)}, \tag{32}$$

$$g_{\text{avg}}^{(b)} = \widetilde{\nabla}_{\theta_{\text{out}}} \widehat{W}_D(P_r, P_s^{\text{avg}}). \tag{33}$$

The implementation differentiates positive generated critic scores. Negating all gradient vectors yields the loss-gradient convention above and leaves the covariance traces and adjacent-step cosine similarities unchanged. At each fixed checkpoint, both networks are in evaluation mode and $B = 6$ latent mini-batches are sampled independently. Each latent vector has 100 standard-normal entries. For either gradient estimator, we use the empirical trace variance $B^{-1} \sum_{b=1}^{B} \|g^{(b)} - \bar{g}\|_2^2$, where $\bar{g} = B^{-1} \sum_b g^{(b)}$, and compute

$$R_{\text{var}} = \frac{\text{Tr}(\text{Cov}(g_{\text{temp}}))}{\text{Tr}(\text{Cov}(g_{\text{avg}})) + \epsilon}. \tag{34}$$

with $\epsilon = 10^{-12}$. A ratio below one indicates lower empirical trace variance for the temporal estimator on the measured parameter subset, as observed at all recorded checkpoints in steps 800–3000 (Table 4). We also compute adjacent-step cosine similarity

$$\rho_t = \frac{\langle g_t, g_{t+1} \rangle}{\|g_t\|_2 \|g_{t+1}\|_2}. \tag{35}$$

The reported cosine similarity averages adjacent step pairs and mini-batches. It describes directional agreement, whereas $\bar{\rho}$ in Proposition 2 measures centered cross-batch covariance over all step pairs. The observed variance reduction supports the proposition's predicted comparison on the final-block parameter subset; testing its sufficient condition additionally requires measuring per-step variances and cross-step covariances. Across the five seeds, the empirical variance advantage coexists with weak, sign-changing adjacent-step cosine similarities.