SQGEN: STRUCTURED QUANTUM IMAGE GENERATION WITH LATENT-MODULATED QUANTIZED TENSOR TRAINS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating images directly from quantum systems is an attractive but unresolved goal on NISQ hardware. Existing quantum generators face several coupled obstacles: barren plateaus that block trainability, expensive quantum circuit preparation, and hardware noise that erodes quantum information with depth. A further difficulty is generating image-scale outputs without a classical decoder. We propose SQGen, a decoder-free quantum generator whose circuit follows a quantized tensor train (QTT) skeleton with latent modulation. Specifically, SQGen promotes the QTT bond index of the target pixel distribution to ancilla bond qubits, so that each circuit site acts on a bond register, the two physical qubits that carry the row- and column-bit of one image scale, and a coarse-to-fine link to the previous scale. The latent enters through latent modulation: an exact angle-level factorization of each re-uploading rotation into a trainable main path plus an additive latent term, reducing to the main path when the latent term is disabled. During training, we create a differentiable model in the classical system under gate-compatibility constraints, with a torus prior as the latent distribution. Every trained operator maps one-to-one to a native quantum gate, yielding a compact, deployable quantum circuit with no classical decoder in the inference path. Extensive experiments on image datasets and synthetic data demonstrate that SQGen trains stably, generates images end-to-end from a shallow circuit with no classical decoder, and shows promising feasibility on real quantum hardware.

1 INTRODUCTION

Quantum generative modeling has emerged as a prominent direction in quantum machine learning, motivated by the prospect that quantum states and circuits offer compact, expressive representations of high-dimensional probability distributions. Parameterized quantum circuits realize this idea through the Born rule, defining distributions that, in certain regimes, are believed to be classically hard to sample (Benedetti et al., 2019; Coyle et al., 2020; Cerezo et al., 2021a); this hardness underlies the long-term prospect of a quantum sampling advantage. The field has since developed several model families: Born machines (Liu & Wang, 2018), quantum generative adversarial networks (Dallaire-Demers & Killoran, 2018), and variational quantum generators (Romero & Aspuru-Guzik, 2021). These early successes establish quantum circuits as expressive distributional priors. However, they have been demonstrated almost entirely on small, low-dimensional targets, leaving the high-dimensional regime, precisely where a sampling advantage would matter, largely untouched.

Scaling quantum generative models to image-scale targets remains substantially harder, and leading approaches fail for distinct reasons. Deep parameterized circuits trained end-to-end can suffer from barren plateaus (McClean et al., 2018; Cerezo et al., 2021b), where gradient variance vanishes exponentially with qubit count and circuit depth, causing optimization to stall before useful structure is learned. Amplitude-encoded priors, which map a continuous latent variable to quantum-state amplitudes, incur expensive state-preparation costs (Cacioppo et al., 2023; Kölle et al., 2024), and the deep preparation circuits they require can further erode the encoded information.

Underlying these methods is a shared failure: image-scale quantum generation demands deep, complex circuits, yet such depth is what NISQ hardware cannot run, as device noise accumulates with

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.

every additional layer (Wang et al., 2021). Hybrid CNN-quantum pipelines (Huang et al., 2021; Chen et al., 2025; Islam et al., 2026) instead produce high-quality image samples with a shallow quantum component, but place the bulk of their modeling capacity in a classical decoder, so the generative capacity resides mainly in the classical component rather than in the quantum circuit.

To address these challenges, we propose a structured circuit architecture with a fully classical training pipeline. First, a QTT bond skeleton is made physical inside the circuit: each bond index is promoted to a small ancilla register whose size scales logarithmically with the bond dimension, and each site applies a local unitary on that register together with the two physical qubits encoding one image scale. The model output is the bond-marginal Born distribution, recovered by sampling all qubits and discarding the ancillas. Second, the latent enters through latent modulation, an instance of data re-uploading (Pérez-Salinas et al., 2020) with fixed integer frequencies, which makes the output exactly 2π-periodic in the latent and a torus the natural latent domain. Finally, training is fully classical: we train a differentiable model whose operators are the native gates of the circuit, with exact backpropagated gradients, which removes shot-based gradient estimation. The trained parameters export one-to-one to a native-gate circuit with no classical decoder. At the evaluated scales the model is efficiently simulable classically; our claims concern structure, trainability, and NISQ deployability rather than a sampling advantage.

We evaluate SQGen on MNIST, Fashion-MNIST, synthetic targets, and real hardware under two protocols: conditional fidelity, measured by per-image KL and L_1 at inferred latents, and generative behavior, probed by images sampled from the deployment prior and by latent-conditional variation. Across our evaluated settings, SQGen trains stably and attains the lowest per-sample KL and L_1 on all ten digit classes while keeping diversity non-zero. The ablation exposes a synergy between the two components: the bond skeleton alone reduces to a static class prototype, latent modulation alone may be harmful without the bond, and only their combination delivers both sharp class structure and latent-conditional variation. Notably, the trained circuit generates end-to-end with no classical decoder, and shows promising feasibility when executed on real quantum hardware. In summary, our main contributions are as follows:

- We propose SQGen, a structured quantum image generator whose circuit follows a multiscale QTT bond skeleton using few qubits, with the latent injected through integer-frequency latent modulation that is exactly periodic on the torus.
- We develop a fully classical training pipeline, where a differentiable gate-compatible model is optimized with exact backpropagated gradients under a torus prior, using a discardable encoder, enabling one-to-one parameter export to a native-gate quantum circuit without a classical decoder at inference.
- We empirically demonstrate the effectiveness of SQGen via extensive experiments, showing that it trains stably without collapse, achieves advanced performance, and deploys promisingly on real quantum hardware.

2 Related Work

Quantum generative models. Many quantum generative models follow the variational quantum algorithm (VQA) paradigm (Cerezo et al., 2021a), in which a parameterized circuit is trained by a classical optimizer, with model families distinguished mainly by their training objective. Quantum circuit Born machines (Liu & Wang, 2018) define implicit generative models through the Born rule, with sampling believed to be classically hard for certain circuit families (Coyle et al., 2020). Quantum generative adversarial networks (Dallaire-Demers & Killoran, 2018; Lloyd & Weedbrook, 2018) replace explicit-divergence training with an adversarial discriminator, while variational quantum generators (Romero & Aspuru-Guzik, 2021) target continuous outputs through amplitude-based constructions. Existing demonstrations in this line are largely concentrated on low-dimensional targets. SQGen differs by targeting image-scale, decoder-free conditional generation with a gate-compatible Born machine trained using exact classical gradients.

Hybrid classical-quantum generators. A common route to visually plausible image samples combines a small quantum component with a classical decoder (Huang et al., 2021; Chen et al., 2025; Islam et al., 2026). Such hybrid pipelines can improve sample quality, but much of the representational capacity resides on the classical side, and the decoder remains in the inference path, so

the pipeline is not end-to-end quantum at deployment. SQGen takes a different design choice: a classical encoder is used only for training and is discarded at deployment. The deployed model is a native-gate quantum circuit whose output pixel distribution is read directly from computational-basis measurements with the bond ancillas marginalized.

Tensor networks and quantum circuits. Tensor network (TN) provides compact low-rank representations for high-dimensional data, and several works compile tensor train or MPS representations into quantum circuits by mapping tensor cores to sequences of unitary blocks (Han et al., 2018; Ran, 2020; Lin et al., 2026). In classical machine learning, TN priors have also been used directly for supervised learning (Stoudenmire & Schwab, 2016) and unsupervised generative modeling (Meiburg et al., 2025), showing that low-rank decompositions can provide useful inductive bias for structured data. SQGen connects these directions by making the QTT bond structure physical inside the quantum circuit. The QTT skeleton fixes the circuit topology, while expressivity is supplied by trainable native-gate rotations rather than by post-hoc compilation of an unconstrained tensor network.

3 BACKGROUND

Quantum circuits and Born machines. A quantum circuit on n qubits acts on a state $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$ by a sequence of unitary gates; we use Dirac notation, so $|i\rangle$ denotes a column basis vector indexed by $i \in \{0,1\}^n$ and $\langle i| = |i\rangle^\dagger$ is its conjugate transpose. We restrict to the native gate set $\{R_y(\theta), R_z(\theta), \mathrm{CNOT}\}$ with Pauli rotations $R_a(\theta) = \exp(-i\theta\,\sigma^a/2)$ for axis $a \in \{y, z\}$. Applying a circuit U to the initial state $|0\rangle^{\otimes n}$ and measuring in the computational basis defines the Born distribution, $p(i) = |\langle i|U|0^{\otimes n}\rangle|^2, i \in \{0,1\}^n$. A parameterized circuit U_θ contains gate angles determined by trainable parameters θ on an input z. Quantum circuit Born machine (QCBM) (Liu & Wang, 2018) learns a target distribution by minimizing a divergence from the Born distribution induced by U_θ. However, deep parameterized circuits suffer from barren plateaus (McClean et al., 2018), where gradient variance can decay exponentially with the number of qubits and circuit depth.

Quantized tensor train. A tensor train (TT) decomposes a function of d discrete indices as a chain of small cores connected by bond indices,

$$T(x_1, \ldots, x_d) \;=\; \sum_{b_1,\ldots,b_{d-1}} \prod_{k=1}^{d} A_k[b_{k-1}, x_k, b_k]\,, \tag{1}$$

with cores $A_k \in \mathbb{R}^{\chi_{k-1}\times|x_k|\times\chi_k}$ and boundaries $b_0 = b_d = 1$. The bond dimension χ controls expressivity: at $\chi = 1$ the TT factorizes across indices, and larger χ covers progressively richer correlations. A quantized tensor train (QTT) (Khoromskij, 2011) first maps a one-dimensional index $X \in \{0, \ldots, 2^L - 1\}$ to its coarse-to-fine binary bits and then applies a TT over those bits, yielding a scale-by-scale representation. For a two-dimensional image, the row and column coordinates each contribute L bits; grouping them by scale yields L levels of two bits each, one row-bit and one column-bit per scale, i.e., factorized as $(2_1 \times 2_2 \times \cdots \times 2_L) \times (2_1 \times 2_2 \times \cdots \times 2_L)$. This binary, scale-by-scale structure is a natural fit for quantum encoding: each scale is carried by two qubits, and the QTT cores supply the local unitaries that act on them.

4 METHOD

SQGen is a Born-machine generator whose circuit is designed around a coarse-to-fine image representation and trained entirely on the classical side. We build it on a latent-modulated quantized tensor train (QTT) skeleton (Figure 1), a multi-scale bond structure whose rotations are conditioned on the latent. Every block is constrained to be unitary, so the trained model maps one-to-one onto a native-gate circuit, a natural fit for quantum deployment.

4.1 IMAGE MAPPING AND NOTATION

An image of side $s = 2^{n_s}$ is represented as a probability distribution over $2n_s$ physical qubits, where n_s is the number of multiresolution scales. We use a Morton (Z-order) layout: qubit $q = 2k$ stores the row bit r_k and qubit $q = 2k + 1$ stores the column bit c_k, with bits indexed from the coarsest scale ($k = 0$) to the finest ($k = n_s - 1$). Thus, each qubit pair (q_{2k}, q_{2k+1}) encodes one spatial

scale and successive pairs refine the location from coarse to fine, as detailed in Section A. The target distribution is defined as $q(r, c) \propto x(r, c)$, where x is the image. A Born machine with parameters θ and latent z outputs

$$p_\theta(i \mid z) \;=\; \left|\langle i \mid U_\theta(z) \mid 0^{\otimes n}\rangle\right|^2 . \tag{2}$$

4.2 Bond-augmented QTT skeleton

Classically, QTT decomposition of $p(r, c)$ takes the form

$$p(r_0, c_0, \ldots, r_{n_s-1}, c_{n_s-1}) \tag{3}$$
$$= \sum_{b_1,\ldots,b_{n_s-1}} \prod_{k=0}^{n_s-1} A_k[b_k,\ (r_k, c_k),\ b_{k+1}] \,,$$

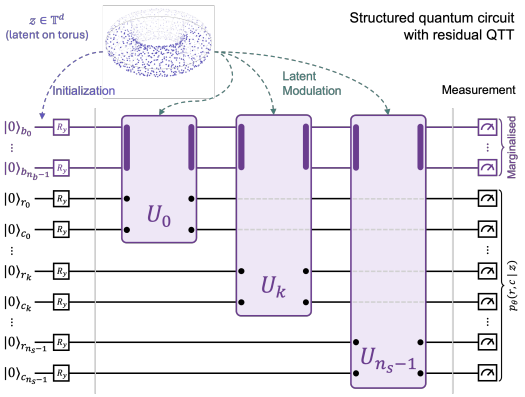


Figure 1: Overview of the SQGen architecture: a latent-modulated quantized tensor train skeleton.

where the bond indices $b_k \in \{1, \ldots, \chi\}$ glue successive sites and $b_0 = b_{n_s} = 1$. Each site A_k is a tensor of shape $(\chi, 4, \chi)$, with the index of size 4 enumerating the joint state $(r_k, c_k) \in \{0, 1\}^2$ of the row and column bits at scale k.

Promotion to qubits. We make this skeleton physical by allocating $n_b = \log_2 \chi$ ancilla bond qubits and replacing each A_k by a unitary block acting on the bond register $\mathcal{H}_b = (\mathbb{C}^2)^{\otimes n_b}$ together with the two physical qubits encoding (r_k, c_k). The full unitary $U_\theta(z) = U_{n_s-1} \cdots U_1 U_0$ is applied to the all-$|+\rangle$ state $|+^{\otimes(n_b+n)}\rangle = \prod_q R_y(\pi/2)\,|0^{\otimes(n_b+n)}\rangle$, and the model output is the bond-marginal Born distribution:

$$p_\theta(r, c \mid z) \;=\; \sum_{b \in \{0,1\}^{n_b}} \left|\langle b, r, c \mid U_\theta(z) \mid +^{\otimes(n_b+n)}\rangle\right|^2 . \tag{4}$$

The image distribution is recovered directly by measuring all qubits in the computational basis and discarding the bond bits, without any classical decoder or post-processing network in inference path.

Local site block. The rotations of site k act on the local register $\mathcal{L}_k = \{q_b : b < n_b\} \cup \{q_{r_k}, q_{c_k}\}$, the n_b bond qubits together with the two physical qubits of scale k. A site is built from L_{site} sub-layers; in each sub-layer ℓ, every local qubit $q \in \mathcal{L}_k$ receives two single-qubit rotations,

$$R_y\left(\alpha^y_{k,\ell,q}(z)\right)\; R_z\left(\alpha^z_{k,\ell,q}(z)\right), \tag{5}$$

followed by a fixed CNOT entangler $E_{k,\ell}$, which for $k \geq 1$ also includes two coarse-to-fine CNOTs between (r_{k-1}, c_{k-1}) and (r_k, c_k), as detailed in Section B. The effective angles $\alpha^a_{k,\ell,q}(z)$ combine trainable parameters with the latent modulation defined in Section 4.3.

Expressivity hierarchy. The bond rank χ provides an explicit capacity parameter for cross-scale dependencies. In the classical skeleton of Eq. (3), setting $\chi = 1$ collapses the decomposition into a product,

$$p(r_0, c_0, \ldots, r_{n_s-1}, c_{n_s-1}) \;=\; \prod_{k=0}^{n_s-1} A_k[(r_k, c_k)] \,, \tag{6}$$

The resulting distribution factorizes across scales, so no dependence between different scale pairs can be mediated through the QTT bonds. Promoting the bond index to an $n_b = \log_2 \chi$-qubit register introduces a shared information channel that is propagated across the n_s circuit sites. Increasing χ enlarges this channel, allowing the circuit to represent richer cross-scale dependencies. The rank therefore defines a hierarchy of structured circuits.

4.3 Latent modulation (LM)

Angle-level modulation. We inject the latent z by refining data re-uploading (Pérez-Salinas et al., 2020; Schuld et al., 2021) at the level of rotation angles. For a single qubit q, axis $a \in \{y, z\}$,

and sub-layer ℓ, the standard re-uploading rotation $R_a(\omega^a_{\ell,q} \cdot z + \beta^a_{\ell,q})$ composed with the trainable rotation $R_a(\theta^a_{\ell,q})$ collapses exactly into a single rotation,

$$R_a\big(\theta^a_{\ell,q}\big)\, R_a\big(\omega^a_{\ell,q} \cdot z + \beta^a_{\ell,q}\big) = R_a\Big(\underbrace{\theta^a_{\ell,q}}_{\text{main path}} + \underbrace{\beta^a_{\ell,q} + \omega^a_{\ell,q} \cdot z}_{\text{latent term}} \Big). \tag{7}$$

This identity follows from the abelian one-parameter group structure of rotations about a common axis. The resulting effective angle at (ℓ, q, a) is

$$\alpha^{a,\,\text{eff}}_{\ell,q}(z) \;=\; \theta^a_{\ell,q} \;+\; \beta^a_{\ell,q} \;+\; \omega^a_{\ell,q} \cdot z. \tag{8}$$

Here, $\theta^a_{\ell,q}$ defines the trainable main path, $\omega^a_{\ell,q} \cdot z$ is the additive latent term, and $\beta^a_{\ell,q}$ is a learnable offset. Restoring the site index k, these effective angles are the $\alpha^a_{k,\ell,q}(z)$ used in Eq. (5). We call this construction latent modulation.

Relation to data re-uploading. Data re-uploading places the data-injection rotation next to each trainable rotation. Latent modulation instead combines the two into a single explicit rotation: by the same-axis identity (Eq. (7)), the trainable and data rotations are exactly equivalent to one rotation whose angle is a trainable main path, a learnable offset, and an additive latent-dependent term. This explicit additive form is what we build on, and it brings two benefits. First, the rewriting is exact, since it follows from an algebraic identity rather than an approximation; moreover, disabling the modulation weights leaves the trainable rotation block intact up to the learnable offset. Second, in deployment, the same-axis rotations can be fused into a single native rotation, reducing the executed single-qubit rotation count. Both formulations realize the same band-limited function class; what differs is the explicitness of the conditioning and the deployed gate cost, as discussed in Section C.

4.4 Classical training, quantum deployment

Unlike previous works, we first train the QTT model in a classical system and then export it one-to-one to a native-gate quantum circuit for deployment. We describe each part in turn.

Latent prior and encoder. Rather than using the isotropic Gaussian prior common in classical generative models (Kingma & Welling, 2013), we place the latent variable z on the torus $\mathbb{T}^d$ and sample it from a scrambled Sobol quasi-Monte Carlo (QMC) sequence. The torus matches the exact 2π periodicity of the generator in z (Section D), so the prior support coincides with the domain on which z acts as a circuit input. The Sobol point set further replaces independent random samples with a low-discrepancy set, giving more uniform coverage of the latent domain and a lower-variance prior estimate at fixed batch size; details are given in Section D. To amortize training over images, a small CNN-based encoder $E_\phi : x \mapsto z \in \mathbb{T}^d$ maps each image to a torus latent, which is matched to the prior by a squared maximum mean discrepancy (MMD) penalty

$$\mathcal{L}_{\text{prior}} = \text{MMD}^2\Big(\Phi(E_\phi[\mathcal{B}]), \Phi(\tilde{\mathcal{Z}})\Big), \tag{9}$$

where $E_\phi[\mathcal{B}] = \{E_\phi(x) : x \in \mathcal{B}\}$, $\mathcal{B}$ is a minibatch of images, $\tilde{\mathcal{Z}}$ is a set of Sobol prior points of the same size, and $\Phi(z) = [\cos z, \sin z]$ respects the 2π-periodic geometry of the torus. The encoder is used only during training and discarded at deployment.

Reconstruction objective. Encoder and QTT model are trained by per-image reconstruction: for each image x, the Born distribution at $z = E_\phi(x)$ is matched to the corresponding target pixel distribution q_x, rather than matching the data distribution at the population level. The reconstruction objective combines a Kullback-Leibler (KL) divergence and an L_1 distance:

$$\mathcal{L}_{\text{rec}}(p_\theta, q_x) = \lambda_{\text{KL}} \text{KL}(q_x \| p_\theta) + \lambda_{L_1} \| p_\theta - q_x \|_1, \tag{10}$$

where λ_{KL} and λ_{L_1} are hyperparameters.

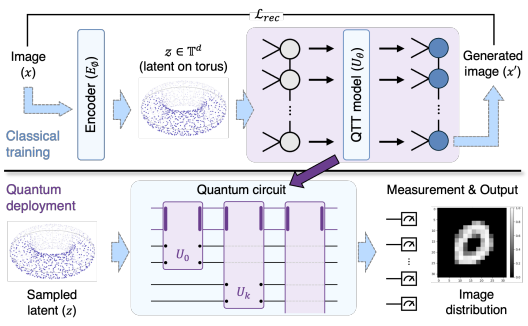


Figure 2: Overview of classical training and quantum deployment pipelines.

Training and deployment pipeline. As shown in Figure 2, training is performed entirely on

the classical system. The encoder E_ϕ and the gate-compatible QTT model U_θ are optimized jointly with AdamW (Loshchilov & Hutter, 2017), where $p_\theta(\cdot \mid z)$ is computed by differentiable tensor contractions and exact gradients are backpropagated through both the encoder and the parameterized QTT tensors. This avoids finite-shot sampling overhead and removes quantum-circuit simulation from the training loop. After training, the encoder is discarded and the learned QTT parameters are exported one-to-one to a native-gate quantum circuit. At deployment, the input of quantum circuits is sampled directly from the torus prior on $\mathbb{T}^d$, the circuit is measured in the computational basis, and the image distribution is obtained by marginalizing out the bond ancillas.

5 EXPERIMENTS

We evaluate SQGen on several tasks. Conditional fidelity asks how accurately the circuit can reproduce images, and is measured per image at inferred latents. Generative behavior asks how the output varies under the deployment protocol, in which latents are drawn from the Sobol torus prior without any encoder, and is probed by prior-sampled images and latent-conditional variation. We additionally deploy SQGen on real quantum hardware to assess practical feasibility.

5.1 SETUP

Datasets. We evaluate SQGen on MNIST (LeCun, 1998), Fashion-MNIST (Xiao et al., 2017) datasets, and a suite of synthetic controlled targets, including stripes and Gaussian blobs. Each image is zero-padded to $2^{n_s} \times 2^{n_s}$ pixel grid matching the physical-qubit budget, and treated as a per-class conditional pixel distribution.

Evaluation metrics. Fidelity metrics are computed at inferred latents: for each x, the circuit is evaluated at $E_\phi(x)$ and its output is compared with the pixel distribution through KL divergence and L_1 distance. Generative behavior is assessed under the deployment protocol: sample grids generated from Sobol prior draws without the encoder, and a diversity score measuring the pairwise output variation across latents.

Baselines. Vanilla BM and BM+LM serve as standard layered QCBM and data re-uploading Born machine baselines (Liu & Wang, 2018; Pérez-Salinas et al., 2020), trained under the same protocol and qubit budget for a matched four-way comparison rather than a purely internal ablation.

Implementation details. The proposed method is trained end-to-end on a differentiable model implemented in PyTorch (Paszke et al., 2019), then the trained model is exported to a quantum circuit in Qiskit (Javadi-Abhari et al., 2024). More details are reported in Section F.

5.2 IMAGE GENERATION ON MNIST DATASETS

Table 1 reports the four configurations alongside two trivial baselines, a class-mean predictor and a uniform distribution. The per-sample errors $\mathrm{KL/x}$ and $\mathrm{L1/x}$ compare the output at the inferred latent $z = E_\phi(x)$ with the target of each test image; $\mathrm{KL_c}$ and $\mathrm{L1_c}$ compare the per-class average output with the empirical class mean; and diversity (Div.) is the mean pairwise L_1 distance between a class's samples. Among the four configurations, SQGen attains the lowest $\mathrm{KL/x}$ and $\mathrm{L1/x}$ and the highest diversity. The static class-mean predictor remains strong on per-sample error, attaining a lower $\mathrm{KL/x}$ than SQGen while SQGen is lower on $\mathrm{L1/x}$: a prototype that never varies scores well on this criterion by construction, and SQGen trades a small amount of per-sample error for latent-conditional variation. The per-digit view in Figure 3 confirms the within-grid comparison is not an averaging artifact, with SQGen winning both metrics on all ten digits (per-digit numbers in Section G.1). Figure 4.(a) shows the same comparison visually, one digit class per column. Importantly, all generated samples in Figure 4.(a) are produced by the deployment protocol itself: the latents are drawn from the Sobol torus prior and the encoder is not used, so the figure directly reflects prior-sampled generation. Results on Fashion-MNIST are reported in Table 5.

Ablation analysis. The four configuration comparison shows that the QTT bond skeleton and latent modulation (LM) are individually insufficient and pay off in combination. With the bond but no modulation, BM+QTT reduces to a sharp but z-invariant prototype whose diversity is zero by construction, so it produces no image-level variation. With modulation but no bond, BM+LM injects z-dependent variation but loses per-sample fidelity, giving the worst $\mathrm{KL/x}$ of the four. Fig-

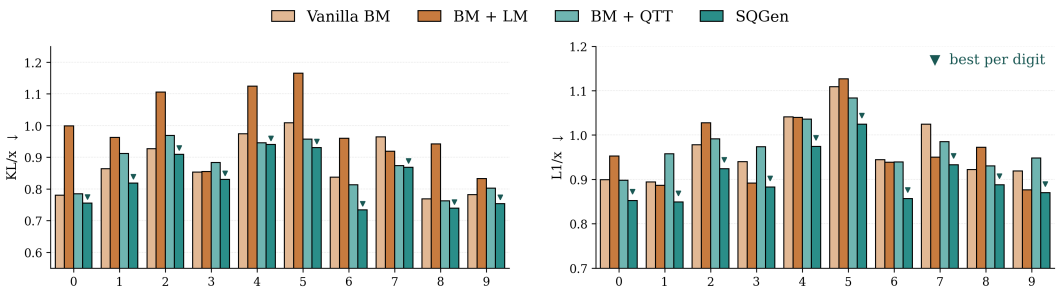


Figure 3: MNIST image generation on per digit. KL/x (left) and L1/x (right) are reported; side-by-side bars compare the four configurations; ▽ marks the best performance.

Method	KL/x ↓	L1/x ↓	KL_c	$L1_c$	Div. ↑
Vanilla BM	$0.876_{\pm 0.088}$	$0.967_{\pm 0.070}$	$0.141_{\pm 0.034}$	$0.299_{\pm 0.046}$	$0_{\pm 0}$
BM + LM	$0.986_{\pm 0.113}$	$0.966_{\pm 0.079}$	$0.282_{\pm 0.073}$	$0.455_{\pm 0.063}$	$0.184_{\pm 0.082}$
BM + QTT	$0.870_{\pm 0.076}$	$0.974_{\pm 0.054}$	$0.129_{\pm 0.042}$	$0.290_{\pm 0.055}$	$0_{\pm 0}$
SQGen	$\mathbf{0.828}_{\pm 0.081}$	$\mathbf{0.905}_{\pm 0.058}$	$0.171_{\pm 0.066}$	$0.348_{\pm 0.059}$	$\mathbf{0.214}_{\pm 0.175}$
Class mean	0.753	0.918	0	0	0
Uniform	2.121	1.719	1.362	1.396	0

Table 1: MNIST image generation, aggregated over 10 digits (mean ± std). The class-mean baseline trivially achieves $KL_c = 0$ and is reported only as a lower-bound reference.

ure 4.(a) shows the effect directly: modulation alone (b→c) yields visibly noisy samples, the bond alone (b→d) recovers a faithful but static prototype, and their combination (e) reproduces the empirical class mean (a) with genuine per-image variation in Table 1. The two mechanisms are thus complementary rather than substitutable, and only SQGen secures class-faithful structure and latent-conditional variation at once. The experiments on Fashion-MNIST dataset also show similar results, reported in Table 5 and Table 6.

5.3 Synthetic controlled targets

We further evaluate SQGen on two controlled targets with known structure. These tasks are not intended as standalone image-generation benchmarks; rather, they provide visually interpretable tests of two behaviors needed by SQGen: representing multi-scale spatial patterns and producing latent-dependent changes in the output distribution.

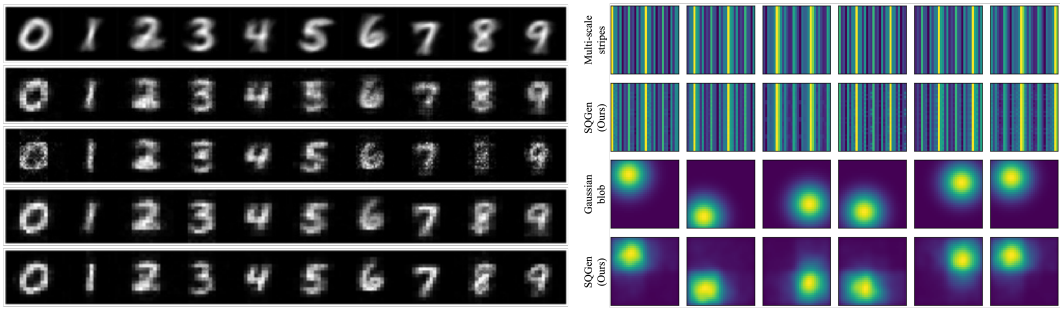
Multi-scale stripes. The first target is a z-invariant multi-scale stripe pattern formed by combining column-wise cosine components at several spatial frequencies. This target provides a simple visual test of whether the model can represent coarse and fine spatial variations simultaneously. As shown in Figure 4.(b), SQGen closely reproduces the stripe structure.

Continuous-position Gaussian blob. The second target is a two-dimensional Gaussian blob whose center changes with the latent variable $z \in \mathbb{T}^2$. This task tests whether the model output varies consistently with a continuous latent input, rather than collapsing to a fixed template. As shown in Figure 4.(b), the generated blobs move with the target locations across Sobol latent draws taken directly from the prior, without the encoder. These controlled examples provide qualitative evidence for the intended behavior of architecture.

5.4 Bond-rank scaling

We sweep $\chi \in \{2, 4, 8, 16\}$ with all other hyperparameters fixed at the defaults of Table 1, reporting MNIST digit 0 in Table 2 and the average over all ten digits in Table 8.

Bond-rank sensitivity. The rank of tensor networks (TN) is the main capacity parameter in TN representation. A larger rank allows the TN model to transmit richer information across scales and



(a) (b)

Figure 4: (a) MNIST samples generated from Sobol prior. From top to bottom are: empirical class mean (target prototype); Vanilla BM; BM+LM; BM+QTT; SQGen. (b) Synthetic targets and SQGen outputs on: a multi-scale cosine sum on the column axis; and 2-D Gaussian blobs on $z \in \mathbb{T}^2$.

Method	χ	KL/x	L1/x	$\mathrm{KL_c}$	$\mathrm{L1_c}$	Method	χ	KL/x	L1/x	$\mathrm{KL_c}$	$\mathrm{L1_c}$
BM+QTT	2	0.813	0.925	0.224	0.416	SQGen	2	1.158	1.031	0.368	0.549
BM+QTT	4	0.787	0.905	0.198	0.386	SQGen	4	0.754	0.853	0.230	0.411
BM+QTT	8	0.785	0.898	0.196	0.360	SQGen	8	0.755	0.852	0.166	0.341
BM+QTT	16	0.799	0.892	0.210	0.360	SQGen	16	0.733	0.841	0.207	0.384

Table 2: The comparison of Bond-rank sweep on the MNIST dataset.

thus represent more detailed spatial structure, but it also increases the bond size and the associated cost. As shown in Table 2, the fidelity of SQGen improves as χ increases from 2 to 4, while the benefit largely saturates at higher ranks. These results are consistent with classical TNs.

Default bond-rank selection. We further repeat the bond-rank sweep across all MNIST digits in Table 8. The aggregate trend is not monotone in χ: the mean at $\chi = 4$ is no better than at $\chi = 2$ and carries the largest spread, while $\chi = 8$ attains the lowest mean KL/x with the smallest cross-digit spread. Diversity moves the opposite way, decreasing as χ grows, since a larger bond rank tightens representation of the class mean at the cost of per-sample variation. We adopt $\chi = 8$ as the default: it gives the lowest mean fidelity error while maintaining non-zero diversity, although the cross-digit spread is comparable to the differences between ranks, so we read the sweep as a practical default choice rather than a statistically separated ranking. This choice also balances the additional expressivity of the bond register against the increased circuit cost of larger ranks.

5.5 Training stability

One motivation for training SQGen on the classical side is to avoid the practical trainability barriers of deep variational quantum circuits, where barren plateaus can flatten gradients and stall optimization (McClean et al., 2018; Cerezo et al., 2021b). In our experiments, SQGen does not exhibit empirical optimization collapse. Figure 5.(a) reports the KL and L_1 training losses averaged over the ten digits. All curves decrease steadily from initialization, without an extended flat phase at the beginning of training, and SQGen reaches the lowest final KL and L_1 losses. This behavior is consistent with the intended role of the classical training. SQGen is optimized as a differentiable QTT model with exact backpropagated gradients, rather than through shot-based estimates of quantum expectation values. As a result, the training loop avoids the shot noise and measurement overhead that can make barren-plateau regimes practically inaccessible in shot-based variational training. Barren plateaus are an asymptotic phenomenon in qubit count and depth, so these observations certify stable optimization at the evaluated scale rather than immunity in the asymptotic regime.

5.6 Noise robustness

We evaluate the circuits under a controlled depolarizing noise setting: after each CNOT gate, we apply a depolarizing channel with rate p, defined in Section H. To separate the injected noise from

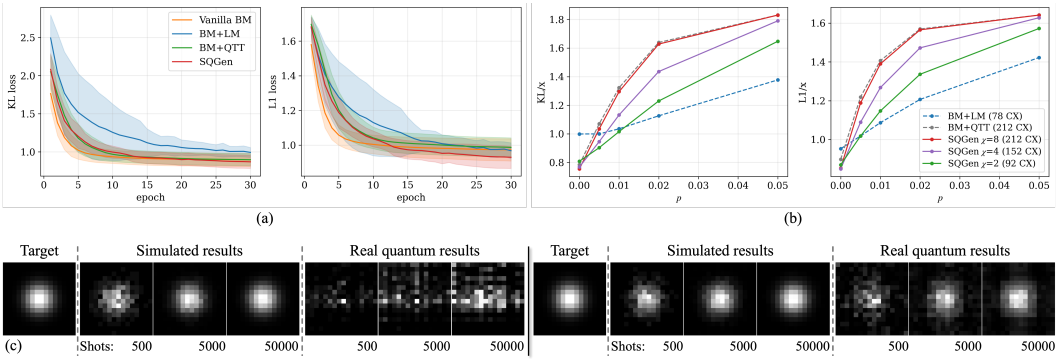


Figure 5: (a) Training loss curves (mean $\pm$ std over the ten MNIST digits): KL (left) and L_1 (right) versus epoch. (b) Noise robustness under Aer simulation. A depolarizing channel is applied after CNOT gate, and we report KL/x (left) and L1/x (right) as functions of the noise rate p. (c) Visualization of AE (left) and SQGen (right) on simulation and real hardware.

shot-based sampling, we compute exact output probabilities with the Aer simulator on digit 0. At $p = 0$, the curves recover the noiseless fidelity, so the degradation as p increases reflects the sensitivity of the circuit to the noise.

All configurations degrade as p increases, and the degradation tracks the CNOT count, since the channel acts after every two-qubit gate. In particular, the default configuration with $\chi = 8$ has the largest bond register among the tested variants and therefore accumulates more injected noise under this model. This illustrates a basic robustness trade-off in the structured circuit: increasing the bond rank improves representational capacity in the noiseless setting, but also increases the number of noisy two-qubit operations. The rank χ provides a direct way to tune this trade-off. Reducing χ decreases the number of bond qubits and CNOT gates, producing flatter degradation curves in Figure 5.(b). The $\chi = 2, 4$ variants are therefore less sensitive to the injected depolarizing noise than the default $\chi = 8$ circuit, although they also provide less cross-scale capacity. Thus, the appropriate rank depends on the desired balance between representational capacity and robustness under the assumed noise level. Overall, this is a controlled sensitivity study rather than a device simulation: current hardware operates at lower per-gate error rates (Section F), and a thorough study of the strong-noise regime is left to future work.

5.7 Real quantum hardware deployment

We further deploy the trained SQGen circuit on real quantum hardware (`ibm_kawasaki` on the IBM Quantum Platform; device details in Section F) and compare it with an amplitude-encoding (AE) baseline, which generates the image through an expensive state-preparation circuit. Figure 5.(c) compares the target images, noiseless simulation results, and real quantum hardware outputs. In the simulation, both SQGen and the AE-based baseline recover the target blob structure reasonably well, indicating that the corresponding ideal circuits encode the desired distributions. However, on real quantum hardware, their behavior differs substantially. SQGen is still able to generate images that closely resemble the target, whereas the AE-based baseline is strongly degraded and no longer retains the useful structure. This contrast suggests that SQGen is more compatible with NISQ hardware, providing promising feasibility in practice.

6 Conclusion

We propose SQGen, a structured quantum generator built on two ideas: a quantized tensor train (QTT) bond skeleton made physical inside the circuit, with integer-frequency latent modulation on a torus latent space; and a classical-training, quantum-deployment pipeline that optimizes a differentiable QTT model before exporting it one-to-one to a native-gate circuit. Across extensive experiments, we empirically demonstrated that SQGen trains stably, attains the best conditional fidelity among four circuit configurations while producing images directly from a shallow circuit with no classical decoder, and shows promising feasibility on real quantum hardware.

AI USE STATEMENT

In this work, we used generative AI tools for language refinement to enhance the clarity and readability of the manuscript. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum science and technology*, 4(4):043001, 2019.

Andrea Cacioppo, Lorenzo Colantonio, Simone Bordoni, and Stefano Giagu. Quantum diffusion models. *arXiv preprint arXiv:2311.15444*, 2023.

Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021a.

Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature communications*, 12(1): 1791, 2021b.

Chi-Sheng Chen, Wei An Hou, Hsiang-Wei Hu, and Zhen-Sheng Cai. Quantum generative models for image generation: Insights from mnist and medmnist. *arXiv preprint arXiv:2504.00034*, 2025.

Brian Coyle, Daniel Mills, Vincent Danos, and Elham Kashefi. The born supremacy: quantum advantage and training of an ising born machine. *npj Quantum Information*, 6(1):60, 2020.

Pierre-Luc Dallaire-Demers and Nathan Killoran. Quantum generative adversarial networks. *arXiv preprint arXiv:1804.08641*, 2018.

Jasmin Frkatovic, Akash Malemath, Ivan Kankeu, Yannick Werner, Matthias Tschöpe, Vitor Fortes Rey, Sungho Suh, Paul Lukowicz, Nikolaos Palaiodimopoulos, and Maximilian Kiefer-Emmanouilidis. On the generalization limits of quantum generative adversarial networks with pure state generators. *Scientific Reports*, 16(1):17845, 2026.

Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised generative modeling using matrix product states. *Physical Review X*, 8(3):031012, 2018.

He-Liang Huang, Yuxuan Du, Ming Gong, Youwei Zhao, Yulin Wu, Chaoyue Wang, Shaowei Li, Futian Liang, Jin Lin, Yu Xu, et al. Experimental quantum generative adversarial networks for image generation. *Physical Review Applied*, 16(2):024051, 2021.

Mujahidul Islam, Serkan Turkeli, and Fatih Ozaydin. Quantum generative adversarial networks: architectures, use cases, and real-world implementations: M. islam et al. *Quantum Information Processing*, 25(1):11, 2026.

Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D Nation, Lev S Bishop, Andrew W Cross, et al. Quantum computing with qiskit. *arXiv preprint arXiv:2405.08810*, 2024.

Boris N Khoromskij. O (d log n)-quantics approximation of n-d tensors in high-dimensional numerical modeling. *Constructive Approximation*, 34(2):257–280, 2011.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Michael Kölle, Gerhard Stenzel, Jonas Stein, Sebastian Zielinski, Björn Ommer, and Claudia Linnhoff-Popien. Quantum denoising diffusion models. In *2024 IEEE International Conference on Quantum Software (QSW)*, pp. 88–98. IEEE, 2024.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Guang Lin, Toshihisa Tanaka, and Qibin Zhao. Structured unitary tensor network representations for circuit-efficient quantum data encoding. *arXiv preprint arXiv:2602.16266*, 2026.

Jin-Guo Liu and Lei Wang. Differentiable learning of quantum circuit born machines. *Physical Review A*, 98(6):062324, 2018.

Seth Lloyd and Christian Weedbrook. Quantum generative adversarial learning. *Physical review letters*, 121(4):040502, 2018.

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

Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):4812, 2018.

Alex Meiburg, Jing Chen, Jacob Miller, Raphaëlle Tihon, Guillaume Rabusseau, and Alejandro Perdomo-Ortiz. Generative learning of continuous data by tensor networks. *SciPost Physics*, 18 (3):096, 2025.

Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020.

Shi-Ju Ran. Encoding of matrix product states into quantum circuits of one-and two-qubit gates. *Physical Review A*, 101(3):032310, 2020.

Jonathan Romero and Alán Aspuru-Guzik. Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions. *Advanced Quantum Technologies*, 4(1): 2000003, 2021.

Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430, 2021.

Edwin Stoudenmire and David Schwab. Supervised learning with tensor networks. *Advances in neural information processing systems*, 29, 2016.

Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature communications*, 12(1):6961, 2021.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

A PIXEL-BASIS MAPPING DETAILS

We use a Morton (Z-order) layout to align image pixels with the QTT site ordering. For an image of side 2^{n_s}, the row and column coordinates are written in binary as

$$r = (r_0, \ldots, r_{n_s-1}), \quad c = (c_0, \ldots, c_{n_s-1}), \tag{11}$$

where r_0, c_0 are the coarsest bits and r_{n_s-1}, c_{n_s-1} are the finest bits. The k-th QTT site is assigned the pair (r_k, c_k), and these two bits are stored on the physical qubits $(2k, 2k+1)$. Thus, each qubit pair represents one spatial scale, and moving along the qubit register refines the pixel location from coarse to fine. Given a computational basis state i, we read its bits as

$$b_{2k}(i) = r_k, \quad b_{2k+1}(i) = c_k, \quad k = 0, \ldots, n_s - 1. \tag{12}$$

The corresponding pixel coordinate is decoded by the usual binary expansion,

$$r(i) = \sum_{k=0}^{n_s-1} r_k \, 2^{n_s-1-k}, \quad c(i) = \sum_{k=0}^{n_s-1} c_k \, 2^{n_s-1-k}. \tag{13}$$

This defines a one-to-one mapping between computational basis states and pixels. Therefore, a Born probability $p_\theta(i \mid z)$ over basis states can be directly interpreted as a pixel distribution $p_\theta(r, c \mid z)$ under this fixed Morton mapping.

Alignment with the QTT sites. The layout matches the QTT structure: the site A_k in Eq. (3) acts on the row-column bit pair (r_k, c_k) that refines the spatial location at scale k. Routing coarse bits to lower qubit indices therefore aligns the qubit ordering with the QTT site ordering, so the circuit follows the same coarse-to-fine organization as the QTT decomposition.

B SITE-BLOCK CONNECTIVITY AND GATE COUNTS

This section specifies the entangler connectivity of the site blocks (Section 4.2) and functions of the bond rank χ, the number of scales n_s, and the per-site sub-layer count L_{site}. Throughout, $n_b = \log_2 \chi$ is the number of bond qubits, $n_{\text{tot}} = n_b + 2n_s$ the total qubit count, and $W_{\text{site}} = n_b + 2$ the width of one local register.

Entangler connectivity. The CNOT pattern $E_{k,\ell}$ of site k at sub-layer ℓ consists of (i) a chain along consecutive bond qubits ($n_b - 1$ CNOTs); (ii) for sites $k \geq 1$, two coarse-to-fine couplings from (r_{k-1}, c_{k-1}) to (r_k, c_k), extending a chain along the physical qubits; (iii) bond-to-physical couplings from every bond qubit to r_k and to c_k ($2n_b$ CNOTs); and (iv) one same-scale coupling $r_k \to c_k$.

Fused rotations. By the same-axis identity in Eq. (7), the trainable and latent-injection rotations on a common axis can be fused into a single rotation per axis. The counts below assume this fused convention; an implementation that keeps the two rotations as separate gates doubles the site-block rotation count and leaves the CNOT count unchanged.

Single-qubit rotations. Each sub-layer applies one R_y and one R_z to every local qubit ($2W_{\text{site}}$ rotations); there are L_{site} sub-layers per site and n_s sites, plus the opening layer $\prod_q R_y(\pi/2)$ on all qubits:

$$\#R_{1q} \;=\; n_{\text{tot}} \;+\; 2\, W_{\text{site}}\, L_{\text{site}}\, n_s. \tag{14}$$

The un-fused implementation instead uses

$$\#R_{1q}^{\text{unfused}} \;=\; n_{\text{tot}} \;+\; 4\, W_{\text{site}}\, L_{\text{site}}\, n_s. \tag{15}$$

Two-qubit gates. Summing the pattern above, site k contributes $3n_b$ CNOTs per sub-layer for $k = 0$ and $3n_b + 2$ for $k \geq 1$, so

$$\#\text{CNOT} \;=\; L_{\text{site}} \big(3\, n_b\, n_s + 2\, (n_s - 1) \big). \tag{16}$$

C LATENT MODULATION AND DATA RE-UPLOADING

This section expands the relation, summarized in Section 4.3, between latent modulation and the data re-uploading primitive it builds on (Pérez-Salinas et al., 2020; Schuld et al., 2021).

Standard data re-uploading. Data re-uploading encodes a continuous input into rotation angles and injects it repeatedly across the circuit: at sub-layer ℓ, qubit q, and axis a, a data-injection rotation $R_a(\omega^a_{\ell,q} \cdot z + \beta^a_{\ell,q})$ is placed next to the trainable rotation $R_a(\theta^a_{\ell,q})$. Its central theoretical property is the Fourier characterization: because z enters the circuit repeatedly, the map $z \mapsto p_\theta(\cdot \mid z)$ is a finite trigonometric series whose frequency support is determined by integer combinations of the weights ω. The primitive specifies repeated input injection, but leaves open how the trainable and data-injection rotations are parameterized and implemented.

Latent modulation. Latent modulation fixes this choice by using the same-axis composition identity. Since rotations about the same axis compose additively, the trainable and data-injection rotations fuse exactly into a single rotation, Eq. (7), whose effective angle in Eq. (8) is the sum of a trainable main path $\theta^a_{\ell,q}$ and an additive latent term $\beta^a_{\ell,q} + \omega^a_{\ell,q} \cdot z$. Two consequences follow. First, the rewriting is an algebraic identity rather than an approximation: relative to the unfused same-axis re-uploading form, it does not change the represented function class, and setting $\omega \equiv 0$ recovers the trainable rotation block up to the learnable offset β. Second, the fused form realizes the latent injection at a lower executed rotation count, quantified in Section B. Both formulations represent the same band-limited function class in z; what changes is the explicitness of the additive conditioning and the deployed gate cost.

Method families	Image scale	Latent cond.	Decoder free	Angle enc.	Structured prior
Born machine (Liu & Wang, 2018)	×	×	✓	✓	×
Quantum GAN (Dallaire-Demers & Killoran, 2018; Lloyd & Weedbrook, 2018)	×	✓	✓	✓	×
Amplitude-encoded (Cacioppo et al., 2023)	×	✓	✓	×	×
Hybrid quantum-classical (Huang et al., 2021; Chen et al., 2025; Islam et al., 2026)	✓	✓	×	✓	×
SQGen (ours)	✓	✓	✓	✓	✓

Table 3: Comparison of representative quantum image-generation families. Columns: Image-scale, demonstrated on image-scale targets rather than only low-dimensional distributions; Latent cond., continuous-latent conditional generation; Decoder-free, no classical decoder in inference; Angle enc., angle-based latent encoding without amplitude state preparation; Structured prior, whether the deployed quantum generator itself incorporates an explicit problem-aligned structural prior.

D TORUS LATENT PRIOR VERSUS A GAUSSIAN PRIOR

Motivation. The latent z enters the circuit through rotation angles Eq. (5), and rotation angles are 2π-periodic: $R_a(\vartheta)$ and $R_a(\vartheta + 2\pi) = -R_a(\vartheta)$ implement the same gate up to a global phase. Since z enters through the effective angle $\alpha(z) = \theta + \beta + \omega \cdot z$ of Eq. (8), periodicity in z additionally requires integer frequencies: every ω is an integer vector fixed at initialization, so a shift $z \mapsto z + 2\pi k$ with $k \in \mathbb{Z}^d$ changes each angle by the multiple $2\pi\,\omega \cdot k$ of 2π and leaves the output distribution unchanged, $p_\theta(\cdot \mid z + 2\pi k) = p_\theta(\cdot \mid z)$. This would not hold for real-valued or trainable frequencies. The prior on z should therefore match the domain on which z acts as a circuit input. We use the torus $\mathbb{T}^d \simeq [-\pi, \pi)^d$, with periodic identification of the boundary.

Comparison with a Gaussian prior. Classical latent-variable generators commonly use an isotropic Gaussian prior on $\mathbb{R}^d$ (Kingma & Welling, 2013). Under this integer-frequency angle encoding, however, a Gaussian prior is many-to-one: latents ϑ and $\vartheta + 2\pi k$, for $k \in \mathbb{Z}^d$, induce the same rotations and therefore the same circuit. Thus, separated regions of Gaussian prior mass can alias to identical quantum inputs. Defining the latent directly on $\mathbb{T}^d$ removes this mismatch by making the latent support coincide with the periodic angle domain.

The encoder is matched to the torus prior through the wrap-respecting embedding $\Phi(z) = [\cos z, \sin z]$ in Eq. (9). For sampling, we replace i.i.d. random draws with a scrambled Sobol quasi-Monte Carlo (QMC) point set on the torus. Sobol points provide more uniform coverage of the latent domain than independent random samples; after fixing the scrambling seed, they also give a reproducible deployment-time latent grid. In our setting, this makes the prior-matching objective more stable and ensures that deployment samples cover the periodic latent domain more evenly.

E Additional related work

Deep parameterized circuits are limited by barren plateaus, in which gradient variance vanishes exponentially with qubit count and depth (McClean et al., 2018; Cerezo et al., 2021b), and hardware noise further flattens the optimization landscape on real devices (Wang et al., 2021). Training Born machines through expectation-style objectives adds a measurement overhead that grows with circuit depth and gradient variance (Cerezo et al., 2021a). In addition, Frkatovic et al. (2026) present a detailed evaluation of quantum generative adversarial networks, reporting that images generated with a 16-qubit circuit and amplitude encoding collapse into noise-like output. Lin et al. (2026) show that, starting from 16×16 images, amplitude-encoding circuits produce almost entirely noise due to deep preparation circuits. Together, these results underscore the challenges of current quantum image generation and motivate the design of SQGen: fully classical training with exact gradients and a structured circuit for quantum image generation.

Capability comparison. Table 3 summarizes representative families of quantum image generators along the design axes discussed in the related work section.

Algorithm 1 SQGen training procedure.

1: Sample minibatch $\{x_i\}_{i=1}^{B}$ and draw Sobol prior points $\{\tilde{z}_i\}_{i=1}^{B} \subset \mathbb{T}^d$.
2: Encode latents $z_i \leftarrow E_\phi(x_i)$ and compute $p_i \leftarrow p_\theta(\cdot \mid z_i)$ by batched differentiable tensor contraction.
3: Compute the reconstruction loss

$$\mathcal{L}_{\text{rec}} = \frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\text{rec}}(p_i, q_{x_i}),$$
$$\mathcal{L}_{\text{rec}}(p_\theta, q_x) = \lambda_{\text{KL}}\,\text{KL}(q_x \| p_\theta) + \lambda_{L_1}\,\|p_\theta - q_x\|_1.$$

4: Compute the prior-matching loss

$$\mathcal{L}_{\text{prior}} = \text{MMD}^2\big(\Phi(\{z_i\}_{i=1}^{B}), \Phi(\{\tilde{z}_i\}_{i=1}^{B})\big)\,.$$

5: Assemble the training objective

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{prior}}\mathcal{L}_{\text{prior}}.$$

6: Backpropagate and take an AdamW step.

F Training and evaluation details

This section expands the setup summarized in Section 5.1.

Training. All configurations are optimized with AdamW (Loshchilov & Hutter, 2017), using QTT model learning rate $\eta_{\text{q}} = 2\times10^{-3}$, encoder learning rate $\eta_\phi = 1\times10^{-3}$, gradient clipping at norm 1.0, and a differentiable forward pass. The batch size is 512 for the plain backend and 256 for the bond backend, and each model is trained for 30 epochs of 50 steps. Algorithm 1 summarizes one training step. The λ_{KL} is 0.5, λ_{L_1} is 1.0, and λ_{prior} is 2.0.

Digit	KL/x	L1/x	$\mathrm{KL_c}$	$\mathrm{L1_c}$
0	0.755	0.852	0.159	0.342
1	0.818	0.849	0.350	0.505
2	0.909	0.924	0.162	0.329
3	0.830	0.883	0.177	0.353
4	0.940	0.974	0.143	0.317
5	0.930	1.024	0.183	0.373
6	0.733	0.857	0.143	0.308
7	0.868	0.933	0.120	0.311
8	0.739	0.888	0.154	0.329
9	0.754	0.870	0.117	0.315
Mean $\pm$ std	0.828 $\pm$0.081	0.905 $\pm$0.058	0.171 $\pm$0.066	0.348 $\pm$0.059

Table 4: Per-digit MNIST metrics for SQGen.

Method	KL/x ↓	L1/x ↓	$\mathrm{KL_c}$	$\mathrm{L1_c}$	Div. ↑
Four configurations					
Vanilla BM	0.399 ± 0.215	0.561 ± 0.204	0.070 ± 0.033	0.177 ± 0.049	0 ± 0
BM + LM	0.446 ± 0.255	0.593 ± 0.219	0.169 ± 0.155	0.317 ± 0.203	**0.199** ± 0.161
BM + QTT	0.428 ± 0.205	0.586 ± 0.191	0.100 ± 0.020	0.221 ± 0.026	0 ± 0
SQGen	**0.394** ± 0.165	**0.550** ± 0.166	0.121 ± 0.036	0.271 ± 0.071	0.175 ± 0.159
Class mean	0.330	0.503	0	0	0
Uniform	1.158	1.297	0.825	1.088	0

Table 5: Fashion-MNIST image generation, aggregated over 10 classes (mean $\pm$ std). The class-mean baseline trivially achieves $\mathrm{KL_c} = 0$ and is reported only as a lower-bound reference.

Latent and deployment. The latent dimension is $d = 4$ and the prior is a scrambled Sobol point set on $\mathbb{T}^4$. After training, the tensors are exported to a native-gate Qiskit circuit whose statevector output agrees with the differentiable QTT model.

Metrics. For each image x, the target q_x is the image normalized to a 1024 distribution under the Morton ordering, and the model output is $p_x = p_\theta(\cdot \mid E_\phi(x))$. We report: Per-sample fidelity: $\mathrm{KL/x} = \mathbb{E}_x[\mathrm{KL}(q_x \,\|\, p_x)]$ and $\mathrm{L1/x} = \mathbb{E}_x[\|p_x - q_x\|_1]$. Class-mean fidelity: $\mathrm{KL_c} = \mathrm{KL}(\bar{q} \,\|\, \mathbb{E}_x[p_x])$ and $\mathrm{L1_c} = \|\mathbb{E}_x[p_x] - \bar{q}\|_1$ against the empirical class mean $\bar{q}$. Diversity: mean pairwise L_1 distance between sampled output distributions.

The tabulated metrics thus quantify conditional fidelity at inferred latents. The qualitative sample grids in the main text are instead produced with the deployment protocol, drawing latents from the scrambled Sobol prior on $\mathbb{T}^d$ without the encoder; quantitative deployment-protocol metrics are reported in Section G.3.

Quantum hardware. The real-hardware experiments of Section 5.7 run on the IBM Quantum Platform; the comparison in Figure 5.(c) uses `ibm_kawasaki` (Heron r2). The backend has 156 qubits and 176 couplers. For the results obtained, the reported median two-qubit (2Q) error was 1.56×10^{-3}, the layered 2Q error was 5.32×10^{-3}, and the best 2Q error was 6.48×10^{-4}. The reported median readout error was 5.98×10^{-3}. The median coherence times were $T_1 = 280.64\,\mu$s and $T_2 = 163.75\,\mu$s. These calibration values provide context for the real-hardware results.

G Additional experiments and discussion

This section reports additional experimental results, including per-digit MNIST metrics, Fashion-MNIST results, and a bond-rank sweep study.

G.1 PER-DIGIT MNIST METRICS

Table 4 breaks the SQGen row of Table 1 down by digit. Digit 4 is the hardest and digit 6 the easiest by KL/x. Digit 1 has the largest class-mean gap, consistent with its concentrated class mean, for which any per-sample variation incurs the steepest class-mean penalty.

G.2 FASHION-MNIST RESULTS

Class	KL/x	L1/x	$\mathrm{KL_c}$	$\mathrm{L1_c}$
T-shirt/top	0.298	0.433	0.113	0.236
Trouser	0.335	0.502	0.123	0.292
Pullover	0.289	0.470	0.105	0.243
Dress	0.315	0.460	0.068	0.191
Coat	0.305	0.435	0.126	0.236
Sandal	0.829	0.984	0.197	0.422
Shirt	0.314	0.469	0.103	0.222
Sneaker	0.350	0.555	0.098	0.242
Bag	0.478	0.643	0.162	0.369
Ankle boot	0.432	0.553	0.121	0.254
Mean	0.394	0.550	0.121	0.271
$\pm$ std	± 0.165	± 0.166	± 0.036	± 0.071

Table 6: Per-class Fashion-MNIST metrics for SQGen.

Table 5 repeats the four-configuration comparison of Section 5.2 on Fashion-MNIST. The architecture and evaluation protocol are unchanged from MNIST. SQGen attains the best per-sample reconstruction of the four configurations. As expected: SQGen is again the only configuration that combines low per-sample error with non-zero diversity. Table 6 gives the per-class breakdown of SQGen metrics.

G.3 GENERATION IN DEPLOYMENT

Method	$\mathrm{Div_{prior}}$	$\mathrm{KL_{prior}}$
BM + LM	1.033 ± 0.006	2.177 ± 0.197
BM + QTT	0	0.419 ± 0.052
SQGen	0.525 ± 0.020	1.689 ± 0.215

Table 7: Deployment-protocol metrics on MNIST, reported as mean $\pm$ standard deviation over the ten digits. For each class, 64 scrambled-Sobol prior samples are evaluated without the encoder.

We further complement the conditional-fidelity metrics with measurements obtained directly under the deployment protocol. For each digit class, the latents are drawn from the scrambled Sobol prior, and the trained circuit is evaluated without the encoder. We report two quantities: $\mathrm{Div_{prior}}$ is the mean pairwise L_1 distance between the prior-sampled output distributions; $\mathrm{KL_{prior}}$ is a nearest-neighbor fidelity metric, computed for each prior sample as the minimum KL divergence to the 64 held-out test images of the same class. Table 7 reports both quantities averaged over the ten digits.

SQGen produces consistently varied outputs under prior sampling, attaining $\mathrm{Div_{prior}} = 0.525 \pm 0.020$ across the ten digits. This confirms that the diversity observed in the main-text sample grids persists under the actual deployment protocol rather than depending on encoder-selected latents. The QTT bond skeleton also improves nearest-neighbor fidelity: SQGen achieves a lower $\mathrm{KL_{prior}}$ than BM+LM (1.689 versus 2.177), consistent with the ablation trend in the main text. In addition, BM+QTT attains the lowest $\mathrm{KL_{prior}}$, but its output is independent of z, so every prior draw produces the same distribution and $\mathrm{Div_{prior}} = 0$. Its low nearest-neighbor divergence therefore reflects the fidelity of a single static prototype rather than successful latent-conditioned generation. Among the latent-dependent models, SQGen provides the balance between prior-sample diversity and proximity to real class examples.

G.4 BOND-RANK SWEEP AGGREGATED OVER MNIST DIGITS

Method	χ	KL/x	L1/x	Div.
BM+QTT	2	0.908 ± 0.086	0.982 ± 0.065	0
BM+QTT	4	0.876 ± 0.080	0.964 ± 0.067	0
BM+QTT	8	0.879 ± 0.078	0.970 ± 0.063	0
BM+QTT	16	0.897 ± 0.089	0.979 ± 0.064	0
SQGen	2	0.927 ± 0.125	0.962 ± 0.079	0.289 ± 0.177
SQGen	4	0.942 ± 0.235	0.987 ± 0.144	0.311 ± 0.236
SQGen	8	0.828 ± 0.081	0.905 ± 0.058	0.214 ± 0.175
SQGen	16	0.923 ± 0.143	0.969 ± 0.091	0.115 ± 0.102

Table 8: Bond-rank sweep aggregated over the ten MNIST digits (mean $\pm$ std across digits).

Table 8 reports the bond-rank sweep of Section 5.4 averaged over the ten MNIST digits, complementing the digit-0 view in Table 2. Mean fidelity is best at $\chi = 8$, while diversity is highest at small χ and decreases as the bond rank grows; this trade-off underlies the default choice $\chi = 8$ discussed in Section 5.4.

H DEPOLARIZING-NOISE MODEL

The noise study in Section 5.6 uses a two-qubit depolarizing channel applied after every CNOT gate. This is a synthetic circuit-noise model; its purpose is to isolate how the circuit output changes as the strength of a simple per-entangling-gate perturbation increases.

Let ρ denote the density matrix before a CNOT gate with ideal unitary U_{CNOT} on two qubits. The ideal update $\rho' = U_{\mathrm{CNOT}}\,\rho\,U_{\mathrm{CNOT}}^{\dagger}$ is followed by the depolarizing channel (Nielsen & Chuang, 2010)

$$\mathcal{D}_p(\rho') \;=\; \Big(1 - \tfrac{15p}{16}\Big)\rho' \;+\; \frac{p}{16}\sum_{P\in\mathcal{P}_2\setminus\{I\otimes I\}} P\rho' P^{\dagger}, \tag{17}$$

where $\mathcal{P}_2 = \{I, X, Y, Z\}^{\otimes 2}$ is the set of two-qubit Pauli operators and $p \in [0, 1]$ is the depolarizing rate. The channel is inserted after every CNOT gate and not after single-qubit rotations, so circuits with more two-qubit gates accumulate more perturbations at the same p. Since the bond circuits contain more CNOTs as χ increases (Section B), this model directly probes the trade-off between cross-scale capacity and sensitivity to two-qubit noise. We evaluate the noisy circuits with exact density-matrix simulation (Qiskit Aer) and read the output probabilities from the diagonal of the final density matrix.