QAMP 2025

Enhancing Quantum Diffusion Models for Complex Image Generation

Jeongbin Jo
jeongbin033@yonsei.ac.kr

December 7, 2025

Contents

---

1 Motivation

1.1 What's the potential benefit of replacing neural network with PQCs?

Parameter Efficiency (Model Compression)

The authors state in their conclusion that ”quantum diffusion models require fewer parameters with respect to their classical counterpart”. For example, their ’conditioned latent’ model, which generated all 10 MNIST digits, used only 1110 parameters. This is a massive reduction compared to the millions of parameters typical in classical U-Net architectures used for diffusion.

Expressivity and Scalability

The paper suggests that for a ”full quantum case,” it is ”possible to generate distributions whose features scale exponentially with the number of qubits”. This implies PQCs have a fundamentally greater expressive power.

Future Potential

The long-term benefit is the potential to ”approximate probability distributions that are not classically tractable”, including ”quantum datasets” themselves.

1.2 How does the proposed QDM compare to just classical DM?

Hybrid Architecture

The paper's QDM is a hybrid model. The forward process (adding noise) is performed classically. The backward process (denoising) is performed by a PQC.

PQC as a Denoising Operator

Instead of having the PQC predict the noise ϵ_θ (like a classical U-Net), this model trains the PQC to be a direct denoising operator $P(\theta, t)$ such that **$P(\theta, t)|x_t\rangle = |x_{t-1}\rangle$**. This avoids the "impractical" cycle of decoding/encoding quantum states at every timestep.

Use of Complex Noise

Because PQC operations can make state coefficients complex, the authors found that using complex Gaussian noise ($\epsilon = \epsilon_r + i\epsilon_i$) during the classical forward process led to a "significant improvement in performance". This is a notable departure from classical models.

Comparable Performance (in Latent Space)

The paper's 'Latent QDM' (which uses a classical autoencoder first) achieved a competitive FID of 38.2 and was able to generate "samples of similar quality to those generated by classical algorithms".

2 Implementation

2.1 Architecture