

“Train classical, deploy quantum” requires rethinking generalization

Snehal Raj^{1,2} Natansh Mathur¹ Alejandro Perdomo-Ortiz¹

¹QC Ware Corp. ²LIP6, Sorbonne Université/CNRS

Abstract

Generative models have become central across science and industry, from image and text synthesis to the design of molecules and materials. Quantum generative models are considered one of the most promising applications for quantum computers, since a quantum circuit naturally produces samples from the distribution it encodes, and for suitable circuits that distribution is believed to be hard for any classical computer to reproduce. A leading strategy trains these models on a classical computer and reserves the quantum device for generating samples at deployment. This is possible when the training loss can be evaluated on a classical computer. A prime example is the maximum mean discrepancy (MMD²), a moment-matching loss that compares the model and the data through their low-order correlations. Research so far has asked whether such models can be trained and whether their sampling is hard; whether minimizing such an objective yields a model that *generalizes*, rather than one that merely reproduces the training statistics, remains poorly understood. We benchmark a broad set of quantum and classical generative models by direct sampling and show that models trained with a moment-matching loss generally show worse generalization than the likelihood-trained models. We show this on two application-inspired datasets: first a cardinality-constrained dataset at up to 30 qubits and second a dataset of genomic single-nucleotide variants, whose valid set is the observed data. These results indicate that a moment-matching loss is not a reliable measure of generalization, and that training quantum generative models for deployment will require approaches that target generalization directly.

1 Introduction

The output of a quantum circuit Born machine is the measurement distribution of a quantum state [1–3], and for suitable circuits that distribution is believed to be hard to sample classically [4, 5]. Training such models on hardware is obstructed by barren plateaus [6], so a growing line of work trains entirely on a classical computer and uses the quantum device only to draw samples at inference [7–9]; this aligns with the observation that the provable absence of barren plateaus tends to coincide with classical simulability of a model’s local observables [10].

Two questions about this program have drawn most of the attention: whether the loss can be optimized [11], and whether sampling is genuinely hard for a classical computer [10, 12]. A third is open: once the loss has converged, does the model *generalize*? Gili et al. [13] distinguish an efficient learner (reproducing the training distribution) from an efficient generator (producing novel valid samples, i.e., generalizing), diagnosed via four metrics: exploration, fidelity, rate, and coverage (EFRC). This framework applies to any discrete distribution for which sample validity can be efficiently checked. Scoring a generative model by the valid and novel samples it produces is long-standing practice in molecular design [14]. A moment-matching objective constrains only low-order moments, which do not pin down the full distribution. We test this on two application-inspired datasets up to 30 qubits with full state-vector simulation, a cardinality-constrained family and a set of genomic single-nucleotide variants. On both datasets, the moment-matching models, in general, generalize worse than the likelihood-trained ones. Across the model zoo, the moment-matching loss also correlates poorly with the forward Kullback-Leibler (KL) divergence, which is the gold standard for estimating generalization given that, for the datasets studied here, the target data distribution is known by design. In addition, we observe a better correlation between coverage and forward KL than with MMD². Therefore, a low MMD² loss is not a reliable sign that a model generalizes (Fig. 1). The companion paper explains theoretically, and verifies numerically, why moment-matching objectives like the MMD² cannot certify generalization.

2 Methods

Datasets. We use two datasets. The first is the cardinality-constrained family [13] which arises widely in many real-world scenarios, including portfolio optimization, feature selection, and signal processing [15].

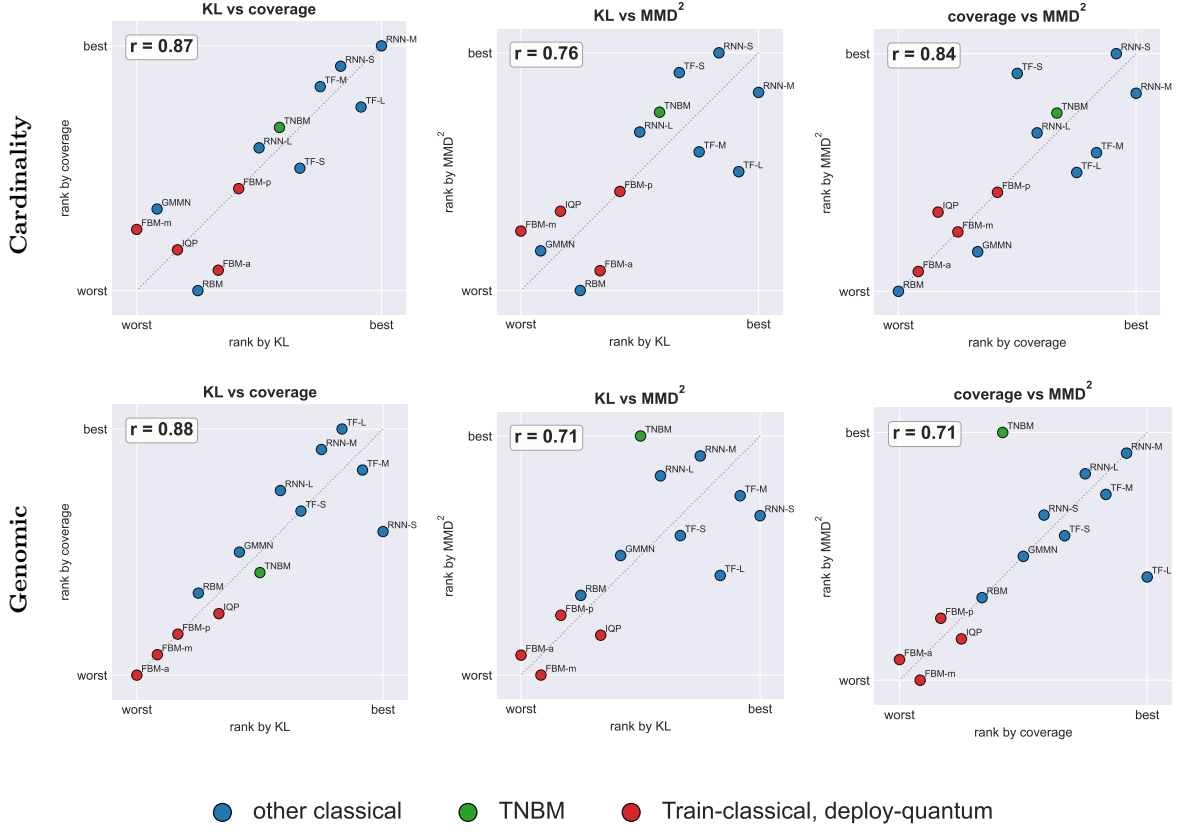


Figure 1: A model’s training loss does not reveal how well it generalizes. Each panel ranks the same thirteen models by two scores, with the best model in the top-right corner, for the cardinality data (top) and the genomic data (bottom); r is the Pearson correlation between the two rankings. Coverage and the forward KL, which both measure generalization, rank the models in close agreement (left column). The moment-matching loss correlates poorly with both (middle and right), so it cannot stand in for a direct test of generalization. In the labels, TF is the transformer and S/M/L are the small, medium, and large variants.

A string of length N is *valid* if exactly k of its bits are one, the valid set is $S = \{x : |x| = k\}$ with $|S| = \binom{N}{k}$, and the target is uniform on S . Without loss of generality, throughout this work, we set $k = N/2$ corresponding to the largest subspace of valid strings for each N . The second is a real-world set of genomic single-nucleotide variants [16]. From the $M = 5,008$ observed sequences we take the first N loci and deduplicate; the resulting distinct variants form the valid set ($|S| = 3,980$ at $N = 20$, 2,715 at $N = 16$), with the target uniform over them. We train each model on a fraction ε of the data.

Generalization metric. A model generalizes when it produces *new, valid* samples: strings that lie in the valid set S but not in its training set T . We measure this by *free sampling*, drawing Q samples from the trained model and keeping all of them. Let G_{sol} be the set of distinct valid strings the model generates that are not in T . The *coverage* $C = |G_{\text{sol}}|/|S \setminus T|$ is the fraction of the unseen valid set that the model actually recovers, and the *fidelity* F is the fraction of its new samples that are valid. Although these metrics have been shown to track generalization, the train-classically program scores models by their loss and correlation matrix, which reflects how well a model matches the data but not whether it generates beyond it. Train-classical, deploy-quantum approaches have so far targeted models that are hard to sample classically, so checking their generalization at scale is costly, unlike classical and quantum-inspired models such as the TNBM.

Model zoo. We benchmark a wide range of classical and quantum generative models: the IQP Born machine [7, 17], the passive (number-conserving), active, and magic fermionic Born machines (FBM) [8],

Table 1: A benchmark of quantum and classical generative models, each scored by both its training loss and its ability to generate new, valid data. Rows group models into likelihood-trained and moment-trained families (Section 2), each trained on a fraction ε of the data and evaluated by free sampling. Two datasets: a cardinality task at $N = 30$ ($\varepsilon \approx 3 \times 10^{-4}$, three seeds) and genomic single-nucleotide variants at $N = 20$ ($\varepsilon = 0.20$, five seeds). We report eval MMD^2 ($\sigma=1$) for both, fidelity F and normalized coverage $\tilde{C} = C/C^*$ for the cardinality columns, and coverage C for the genomic. More details on the models and labels is given in the model zoo (Section 2).

Model	loss	Cardinality ($N=30$)			Genomic ($N=20$)	
		MMD ²	F	$\tilde{C}$	MMD ²	C
<i>Likelihood-trained</i>						
Transformer (L)	cross-ent	1.00e-3	1.00	1.00	1.72e-3	0.134
Transformer (M)	cross-ent	1.00e-3	1.00	1.00	1.39e-3	0.139
RNN (S)	cross-ent	1.02e-3	1.00	1.00	3.21e-3	0.083
TNBM	NLL	1.01e-3	0.96	0.96	8.00e-4	0.141
RNN (M)	cross-ent	1.02e-3	0.85	0.85	1.09e-3	0.138
RNN (L)	cross-ent	1.01e-3	0.91	0.91	1.24e-3	0.134
Transformer (S)	cross-ent	9.97e-4	0.71	0.71	2.91e-3	0.090
RBM	CD	1.04e-3	0.15	0.15	5.67e-3	0.025
<i>Moment-trained</i>						
FBM (passive)	Z-corr	1.02e-3	1.00	1.00	6.59e-3	0.029
magic FBM	MMD ²	1.09e-3	0.16	0.16	2.44e-2	0.001
GMMN (classical MMD)	MMD ²	1.72e-3	0.24	0.24	2.01e-3	0.129
IQP	MMD ²	1.43e-3	0.14	0.14	1.08e-2	0.012
FBM (active)	Z-corr	1.39e-3	0.09	0.09	2.10e-2	0.000

a tensor-network Born machine (TNBM) [18], autoregressive transformers (TF) and RNNs each in small, medium, and large variants (S/M/L), a restricted Boltzmann machine (RBM), and a generative moment-matching network (GMMN) [19, 20]. The transformers, RNNs, TNBM, and RBM are likelihood-trained; the IQP, the active and magic FBM, and the GMMN are moment-trained. The passive FBM also uses a moment loss, but its circuit emits only fixed-weight strings, confining it to the cardinality valid set by construction.

3 Results

Cardinality-constrained data. We train each model with its own loss and, after training, measure both its MMD^2 and its coverage. On this task loss and coverage decouple: the deploy-quantum moment-trained models (the IQP Born machine and the active and magic FBM) show convergence of the MMD^2 to a low value, yet these do not necessarily efficiently generate bitstrings from the valid set. A low loss does not make them generators. The likelihood-trained models behave differently, reaching a comparably low MMD^2 and also covering most of the valid set (Table 1). One moment-trained model is an exception. The passive number-conserving fermionic Born machine reaches high coverage because its matchgate circuit only produces Hamming-weight-preserving strings, so it is confined to the valid set by construction rather than by its loss. Although no other model was biased, our results indicate the importance of a strong inductive bias, when possible, for enhancing a model’s performance.

Genomic variant data. On the genomic data the deploy-quantum models are again the weakest generators. The transformer and the RNN reach the highest coverage, with the tensor-network Born machine close behind, while the deploy-quantum models show poor generalization. Here, those models also fail to reach a low MMD^2 : with no fixed Hamming weight to exploit, they match neither the data’s low-order statistics nor its support. Here loss and coverage seem to have a slightly better correlation than that of the cardinality data set. Whether the MMD^2 tracks generalization thus depends on the dataset, hence reflecting the importance of properly evaluating generalization (Fig. 1). Although the passive fermionic born machine showed good coverage for the cardinality constrained task, for the genomic dataset they show poor generalization similar to other moment-objective trained models.

References

- [1] Marcello Benedetti, Delfina Garcia-Pintos, Oscar Perdomo, Vicente Leyton-Ortega, Yunseong Nam, and Alejandro Perdomo-Ortiz. A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Information*, 5:45, 2019.
- [2] Jin-Guo Liu and Lei Wang. Differentiable learning of quantum circuit born machines. *Phys. Rev. A*, 98:062324, 2018.
- [3] Brian Coyle, Daniel Mills, Vincent Danos, and Elham Kashefi. The born supremacy: Quantum advantage and training of an ising born machine. *npj Quantum Information*, 6:60, 2020.
- [4] Michael J. Bremner, Richard Jozsa, and Dan J. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proc. R. Soc. A*, 467:459–472, 2011.
- [5] Michał Oszmaniec, Ninnat Dangniam, Mauro E. S. Morales, and Zoltán Zimborás. Fermion sampling: A robust quantum computational advantage scheme using fermionic linear optics and magic input states. *PRX Quantum*, 3:020328, 2022.
- [6] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9:4812, 2018.
- [7] Erik Recio-Armengol, Shahnawaz Ahmed, and Joseph Bowles. Train on classical, deploy on quantum: Scaling generative quantum machine learning to a thousand qubits. *arXiv preprint arXiv:2503.02934*, 2025.
- [8] Bence Bakó, Zoltán Kolarovszki, and Zoltán Zimborás. Fermionic born machines. *arXiv preprint arXiv:2511.13844*, 2025.
- [9] Manuel S. Rudolph, Jacob Miller, Danial Motlagh, Jing Chen, Atithi Acharya, and Alejandro Perdomo-Ortiz. Synergistic pretraining of parametrized quantum circuits via tensor networks. *Nature Communications*, 14:8367, 2023.
- [10] M. Cerezo, Martín Larocca, Diego García-Martín, N. L. Diaz, Paolo Braccia, Enrico Fontana, Manuel S. Rudolph, et al. Does provable absence of barren plateaus imply classical simulability? *arXiv preprint arXiv:2312.09121*, 2023.
- [11] Sacha Lerch, Joseph Bowles, Ricard Puig, Erik Armengol, Zoë Holmes, and Supanut Thanasilp. IQP born machines under data-dependent and agnostic initialization strategies. *arXiv preprint arXiv:2603.14576*, 2026.
- [12] Mario Herrero-González, Brian Coyle, Kieran McDowall, Ross Grassie, Sjoerd Beentjes, Ava Khamseh, and Elham Kashefi. The born ultimatum: Conditions for classical surrogation of quantum generative models with correlators. *arXiv preprint arXiv:2511.01845*, 2025.
- [13] Kaitlin Gili, Marta Mauri, and Alejandro Perdomo-Ortiz. Generalization metrics for practical quantum advantage in generative models. *Phys. Rev. Applied*, 21:044032, 2024. arXiv:2201.08770.
- [14] Nathan Brown, Marco Fiscato, Marwin H. S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking models for de novo molecular design. *J. Chem. Inf. Model.*, 59(3):1096–1108, 2019.
- [15] Andreas M Tillmann, Daniel Bienstock, Andrea Lodi, and Alexandra Schwartz. Cardinality minimization, constraints, and regularization: a survey. *SIAM Review*, 66(3):403–477, 2024.

- [16] Burak Yelmen, Aurélien Decelle, Linda Ongaro, Davide Marnetto, Corentin Tallec, Francesco Montinaro, Cyril Furtlehner, Luca Pagani, and Flora Jay. Creating artificial human genomes using generative neural networks. *PLoS Genetics*, 17(2):e1009303, 2021.
- [17] Maarten van den Nest. Classical simulation of quantum computation, the gottesman-knill theorem, and slightly beyond. *Quantum Inf. Comput.*, 10:258, 2010.
- [18] Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised generative modeling using matrix product states. *Phys. Rev. X*, 8:031012, 2018.
- [19] Yujia Li, Kevin Swersky, and Richard Zemel. Generative moment matching networks. In *Proc. 32nd International Conference on Machine Learning (ICML)*, pages 1718–1727, 2015.
- [20] Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Proc. 31st Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 258–267, 2015.