GENERATIVE MODEL VIA QUANTILE ASSIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning complex high-dimensional distributions is a challenging problem, and modern generative models address it through a variety of architectures and training objectives, often involving auxiliary learned components such as encoders, discriminators, or critics. We introduce NEUROSQL, a generative modeling approach whose only learned component is a single neural generator. NEUROSQL fixes a lattice of multivariate quantile points in latent space and trains the generator by alternating between a regression step and an assignment step that matches data points to latent quantiles. Under squared Euclidean cost, exact assignment yields the optimal transport coupling between the empirical data measure and the generator's pushforward of the quantile lattice, so the training objective is exactly their squared Wasserstein-2 distance. This explicit objective makes the optimization amenable to analysis: under suitable smoothness and step-size conditions, we prove that the alternating procedure decreases the objective monotonically and reaches a stationary point of it at rate $O(T^{-1/2})$ — a guarantee unavailable to adversarial training, which has no single potential function, or to variational training, which is monotone only on a bound. The resulting algorithm is lightweight, fast, and deterministic given the lattice. Controlled two-dimensional experiments reach held-out sliced Wasserstein distances comparable to an independent real–real reference. A million-observation simulation confirms that block assignment keeps cost-matrix memory bounded and solver cost linear in n. When trained on all 60,000 MNIST training images using block assignment, NEUROSQLrecovers all ten classes with no exact training-image copies, and is competitive with matched-backbone VAE and GLO baselines at lower training cost. Our guarantees are finite-sample optimization guarantees rather than population-consistency results, and within-class coverage remains the method's clearest limitation.

1 INTRODUCTION

Deep generative models (DGMs) are now standard tools in machine learning and, increasingly, in scientific fields such as genomics and neuroimaging, where they are used to interpret data, decode images and sequences, and generate realistic datasets. Most latent-variable DGMs pair the generator with an auxiliary network: an encoder in variational autoencoders (VAEs; Kingma & Welling, 2014), a discriminator in generative adversarial networks (GANs; Goodfellow et al., 2014), or a denoiser iterated at sampling time in diffusion models (Ho et al., 2020).

These auxiliary components come at a cost. Practically, training several networks jointly can be unstable: GANs are prone to mode collapse and non-convergence (Mescheder et al., 2018), and VAEs to blurred samples. Diffusion models, in turn, are expensive to train and to sample from, which limits them in resource-constrained or low-data regimes. Theoretically, encoders and discriminators are functions of the high-dimensional observations, whereas the generator takes low-dimensional latent inputs, so they may suffer from a curse of dimensionality (Stone, 1985). The favorable approximation properties of DNNs (Schmidt-Hieber, 2020; Suzuki, 2019) rely on assumptions that DGMs rarely satisfy (Golestaneh et al., 2025), and theoretical analyses of VAEs and GANs often sidestep these difficulties (see e.g. Bodelet et al. 2026; Chae et al. 2023; Biau et al. 2020)

We propose NEUROSQL, a latent-variable DGM whose only learned component is the generator. NEUROSQL fixes a lattice of multivariate quantiles of the prior, the center-outward quantiles of Hallin et al. (2021), and represents the unknown latent realizations as a permutation of this lattice, an approximation that becomes exact as $n \to \infty$. Training alternates between fitting the generator by

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.

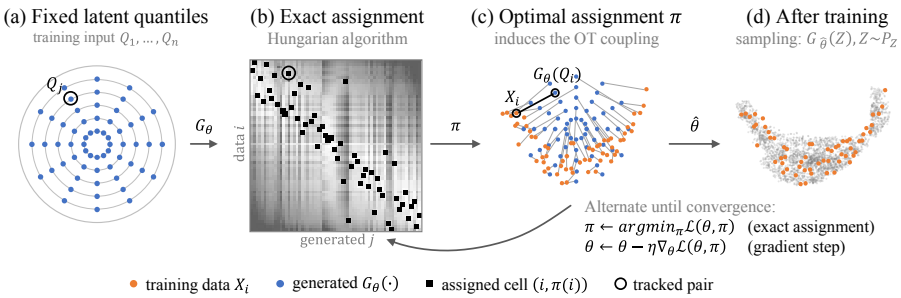

Figure 1: **A schematic representation of the NEUROSQLarchitecture.** A deterministic lattice of latent quantiles $Q_{1:n}$ (blue) is mapped through the generator and matched bijectively to the observations (orange) by solving a linear assignment problem with cost matrix $C_{ij} = \ell(X_i, G_\theta(Q_j))$.

regression and re-solving the permutation as a linear assignment problem. NEUROSQL generalizes Statistical Quantile Learning (Bodelet et al., 2026), which was restricted to additive models and univariate quantiles. Its closest encoder- and discriminator-free precedents are GLO (Bojanowski et al., 2018), which optimizes free latent codes, and IMLE (Li & Malik, 2018), which uses one-sided nearest-neighbour matching. NEUROSQL instead matches data to a fixed lattice bijectively, which gives its objective a distributional meaning (Appendix F discusses further related work). Our contributions are:

- **An exact Wasserstein objective.** Since data and lattice are uniform measures on n atoms, the assignment step solves the optimal transport problem exactly, and the training objective equals the squared Wasserstein-2 distance between the data and the generator's pushforward of the lattice. It is not a bound (VAEs), an adversarial approximation of a dual (WGAN), or an entropically smoothed cost.
- **Optimization guarantees.** NEUROSQL is subgradient descent on this distance: every step is non-increasing, and the iterates reach an $O(T^{-1/2})$-stationary point. To our knowledge, this is the first non-asymptotic convergence guarantee for training a deep generator on the exact Wasserstein distance; existing results concern sliced losses and are asymptotic (Tanguy, 2023).
- **A resource-frugal design.** A single network is trained, no function of the high-dimensional observations is learned, and block assignment makes the solver cost linear in n.
- **Empirical validation** on simulations with up to one million observations, the full MNIST training set, and CelebA, against VAE and GLO baselines (Section 4).

2 QUANTILE ASSIGNMENT

2.1 BACKGROUND

Let's consider a dataset $\mathcal{D}$ composed of n independent samples of p-dimensional random vectors, $\mathcal{D} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}$. We assume that the data are driven by unobserved continuous latent variables $\boldsymbol{Z}_i \in \mathcal{Z} \subseteq \mathbb{R}^d$. Specifically, we consider a probabilistic generative model:

$$\boldsymbol{X}_i = \boldsymbol{G}(\boldsymbol{Z}_i) + \boldsymbol{\epsilon}_i, \tag{1}$$

where $\boldsymbol{G} : \mathcal{Z} \to \mathbb{R}^p$ is an unknown generator, and $\boldsymbol{\epsilon}_i$ are independent random errors. We assume that the latent variables $\boldsymbol{Z}_i$ follow a known continuous distribution P_Z. The latent dimension is typically (much) smaller than the ambient dimension ($d << p$) to enable dimensionality reduction. This, therefore, contrasts with flow-based generative models, where the generator must be invertible. As DGMs are not identifiable, one can select any distribution as long as it has a continuous cumulative distribution function. Regarding the prior distribution, it is common to use either the standard normal distribution, $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, or a Uniform distribution. We approximate the generator using deep neural networks (DNN), that is $\boldsymbol{G} \approx \boldsymbol{G_\theta}$, where:

$$\boldsymbol{G_\theta} \in \{W_L \circ \sigma \circ W_{L-1} \circ \sigma \cdots \circ \sigma \circ W_1\},$$

and W_l denote affine transformations and σ is an activation function. The vector $\boldsymbol{\theta} \in \Theta$ contains the DNN parameters. We aim to learn both the generator parameter $\boldsymbol{\theta}$ and the latent variables $\boldsymbol{Z}_i$.

2.2 Latent space approximation

For deep generative models, likelihood-based estimation is tractable only in simple settings such as linear factor analysis, and jointly learning the generator and the latent variables is computationally demanding. NEUROSQL follows the spirit of the sieve method, which replaces an intractable optimization over a complex parameter space by tractable problems over simpler subspaces that grow dense in the original space (see, e.g., Chen 2007). Here the simplification is on the latent side: the unknown latent realizations are replaced by n quantiles $\boldsymbol{Q}_1^n, \ldots, \boldsymbol{Q}_n^n \in \mathcal{Z}$ that discretize P_Z. The key idea of NEUROSQL is that there exists a permutation π such that $\boldsymbol{Z}_i \approx \boldsymbol{Q}_{\pi(i)}^n$, so that learning the latent variables reduces to learning π.

Formally, we call quantiles any sequence $\boldsymbol{Q}_1^n, \ldots, \boldsymbol{Q}_n^n \in \mathcal{Z}$ satisfying

$$\min_{\pi \in S_n} \frac{1}{n} \sum_{i=1}^{n} \left\| \boldsymbol{Z}_i - \boldsymbol{Q}_{\pi(i)}^n \right\|^2 \xrightarrow{p} 0 \qquad \text{as } n \to \infty, \tag{2}$$

where S_n denotes the symmetric group of order n. The left-hand side is the squared Wasserstein-2 distance between the empirical distribution of the latent variables and the uniform distribution on the quantiles, so equation 2 requires the quantiles to approximate the latent sample in W_2.

Univariate case. For $d = 1$, let $\boldsymbol{Q}_i^n = F_Z^{-1}\big(\frac{i}{n+1}\big)$, $i = 1, \ldots, n$, where F_Z is the cumulative distribution function of P_Z. In one dimension the optimal assignment is monotone: it matches the i-th order statistic $\boldsymbol{Z}_{(i)}$ to $\boldsymbol{Q}_i^n$. The left-hand side of equation 2 therefore equals $\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{Z}_{(i)} - \boldsymbol{Q}_i^n)^2$, which vanishes whenever P_Z has a finite second moment.

Multivariate case. Since $\mathbb{R}^d$ has no canonical ordering, there is no universal definition of multivariate quantiles. We use the center-outward quantiles of Hallin et al. (2021), constructed by optimal transport from a grid in the unit ball; in this paper, we used a Sobol low-discrepancy sequence mapped to the ball. Section A in the Appendix shows that these quantiles satisfy equation 2, and in fact almost surely.

2.3 Training algorithm

Since the unobserved latent realizations can be approximated by the fixed quantiles up to an unknown permutation, i.e. $\boldsymbol{Z}_i \approx \boldsymbol{Q}_{\pi(i)}$ for some $\pi \in S_n$, fitting the generator therefore becomes a regression of $\boldsymbol{X}_i$ on $\boldsymbol{Q}_{\pi(i)}$ in which the permutation is unknown and is estimated jointly with $\boldsymbol{\theta}$. For a nonnegative cost $\ell : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}_+$, we define the training objective

$$\mathcal{L}(\boldsymbol{\theta}, \pi) := \frac{1}{n} \sum_{i=1}^{n} \ell\big(\boldsymbol{X}_i,\, G_{\boldsymbol{\theta}}(\boldsymbol{Q}_{\pi(i)})\big), \tag{3}$$

and define NEUROSQL as a minimizer

$$(\hat{\boldsymbol{\theta}}, \hat{\pi}) \in \operatorname*{arg\,min}_{\boldsymbol{\theta} \in \Theta,\ \pi \in S_n} \mathcal{L}(\boldsymbol{\theta}, \pi), \tag{4}$$

with latent estimates $\hat{\boldsymbol{Z}}_i = \boldsymbol{Q}_{\hat{\pi}(i)}$.

Problem equation 4 couples a continuous parameter with a combinatorial one, since S_n has $n!$ elements, but each block is easy on its own. For fixed $\boldsymbol{\theta}$, minimizing over π is a linear assignment problem with cost matrix $C_{ij} = \ell(\boldsymbol{X}_i, G_{\boldsymbol{\theta}}(\boldsymbol{Q}_j))$. It is solved exactly in $O(n^3)$ time, for instance by the Hungarian algorithm (Kuhn, 1955); any exact solver can be used. For fixed π, equation 3 is a standard regression loss in $\boldsymbol{\theta}$, which we decrease by gradient steps. We alternate these two step (see Algorithm 1):

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta\, \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^t, \pi^t), \qquad \pi^{t+1} \in \operatorname*{arg\,min}_{\pi \in S_n} \mathcal{L}(\boldsymbol{\theta}^{t+1}, \pi). \tag{5}$$

for $t \geq 0$. Both steps decrease the same objective, which is the basis of the analysis in Section 3. In practice, several generator updates (for instance epochs of minibatch stochastic gradient descent) are performed between reassignments, and for large n exact assignment is replaced by block assignment (Section 5).

Algorithm 1 NEUROSQL training

Require: data $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$; quantiles $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_n$; initial parameters $\boldsymbol{\theta}^0$; step size η; number of rounds T

$\pi^0 \leftarrow \arg\min_{\pi \in S_n} \mathcal{L}(\boldsymbol{\theta}^0, \pi)$ ▷ assignment step
for $t = 0, \ldots, T-1$ **do**
  $\boldsymbol{\theta}^{t+1} \leftarrow \boldsymbol{\theta}^t - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^t, \pi^t)$ ▷ generator step
  $C_{ij} \leftarrow \ell\big(\boldsymbol{X}_i, G_{\boldsymbol{\theta}^{t+1}}(\boldsymbol{Q}_j)\big)$ for all i, j
  $\pi^{t+1} \leftarrow \arg\min_{\pi \in S_n} \sum_i C_{i\pi(i)}$ ▷ assignment step (e.g. Hungarian)
end for
return $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^T$ and $\hat{\boldsymbol{Z}}_i = \boldsymbol{Q}_{\pi^T(i)}$, $i = 1, \ldots, n$

Sampling. Training uses only the fixed quantiles $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_n$. New samples are generated as $G_{\hat{\boldsymbol{\theta}}}(\boldsymbol{Z})$ with $\boldsymbol{Z} \sim P_Z$, which is justified by the convergence of the quantile lattice to P_Z as $n \to \infty$ (Proposition 3).

3 THEORETICAL ANALYSIS

We note that all results extend to a penalized objective. For our theoretical analysis, we will only require one standard hypothesis in neural networks literature.

Assumption 1 (Smoothness)**.** *For every fixed π, the map $\boldsymbol{\theta} \mapsto \mathcal{L}(\boldsymbol{\theta}, \pi)$ is differentiable with β_π-Lipschitz gradient. Namely, we assume that*

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \pi) - \nabla_{\boldsymbol{\theta}} \mathcal{L}(\tilde{\boldsymbol{\theta}}, \pi)\| \leq \beta_\pi \|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|,$$

for any $\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}} \in \Theta$. As S_n is finite, $\beta := \max_\pi \beta_\pi$ is finite.

3.1 THE TRAINING OBJECTIVE IS AN EMPIRICAL WASSERSTEIN DISTANCE

We show that our algorithm directly minimizes (up to a penalty) the Wasserstein distance between the data distribution and the assumed model distribution.

Lemma 1 (Wasserstein distance)**.** *Consider the training objective equation 3 with squared loss $\ell(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$. Denote by δ_z the Dirac delta distribution with point mass at z. Let $\hat{\nu}_n = \frac{1}{n}\sum_{i=1}^n \delta_{X_i}$ be the empirical distribution of the data, $\hat{\nu}_n := \frac{1}{n}\sum_{i=1}^n \delta_{\boldsymbol{Q}_i}$ be the distribution of the quantiles. We denote the push-forward by $\boldsymbol{G}_{\boldsymbol{\theta}\#}\hat{\nu}_n = \frac{1}{n}\sum_{i=1}^n \delta_{\boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{Q}_i)}$. For every $\boldsymbol{\theta} \in \Theta$, we have*

$$\min_{\pi \in S_n} \mathcal{L}(\boldsymbol{\theta}, \pi) \;=\; W_2^2\big(\hat{\nu}_n,\; \boldsymbol{G}_{\boldsymbol{\theta}\#}\hat{\nu}_n\big).$$

Lemma 1 thus allows us to characterize the objective as a Wasserstein distance, $\bar{\mathcal{L}}(\boldsymbol{\theta}) := \min_{\pi \in S_n} \mathcal{L}(\boldsymbol{\theta}, \pi)$, where $\hat{\nu}_n \to \nu$, ν denoting the true distribution of the latent variable.

The assignment step is thus not an auxiliary heuristic: it evaluates W_2^2 exactly, and the permutation it returns is the optimal transport plan at the current parameters.

3.2 MONOTONE DESCENT

The concern with alternating schemes is that the two steps may pursue different criteria, so that neither quantity improves and the iterates cycle. This cannot occur here: both steps minimize $\mathcal{L}$ over their respective block.

Proposition 1 (Monotone descent)**.** *Under Assumption 1 and $0 < \eta \leq 1/\beta$, the training objective never increases along the algorithm:*

$$\mathcal{L}(\boldsymbol{\theta}^{t+1}, \pi^{t+1}) \leq \mathcal{L}(\boldsymbol{\theta}^{t+1}, \pi^t) \leq \mathcal{L}(\boldsymbol{\theta}^t, \pi^t) - \frac{\eta}{2}\|\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^t, \pi^t)\|^2.$$

Since $\mathcal{L} \geq 0$, the sequence converges.

In particular, the descent is strict in the case where $\|\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \pi^t)\| > 0$. Namely, neither half of the alternation can undo the other. The generator step decreases the objective by the descent Lemma, and the reassignment step cannot increase it, as the previous assignment remains available to the solver and the solver is exact.

3.3 Stationarity rate

Optimizing π out leaves $\bar{\mathcal{L}}$, a minimum of finitely many smooth functions. Such a function is locally Lipschitz but not differentiable where the optimal assignment changes, and a descent gradient Lemma does not readily apply. However, we will show that we can overcome this difficulty using the Clarke generalized gradient. For completeness, we add the definition of Clarke gradient Clarke (1990):

Definition 1 (Clarke generalized gradient)**.** *Let $f : \Theta \to \mathbb{R}$ be locally Lipschitz. By Rademacher's theorem f is differentiable outside a Lebesgue-null set Ω_f, and the Clarke generalized gradient of f at $\boldsymbol{\theta}$ is:*

$$\partial f(\boldsymbol{\theta}) = \operatorname{conv}\Big\{ v \in \mathbb{R}^{dim(\Theta)} \ : \ \boldsymbol{\theta}_m \to \boldsymbol{\theta},\ \boldsymbol{\theta}_m \notin \Omega_f,\ \lim_{m\to\infty} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_m) = v \Big\},$$

where conv(S) represents the convex hull of the set S. The set $\partial f(\boldsymbol{\theta})$ is nonempty, convex, and compact, and reduces to $\{\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})\}$ whenever f is continuously differentiable at $\boldsymbol{\theta}$.

We can now specify the Clarke gradient of the reduced objective.

Lemma 2 (Subgradient descent)**.** *Define the set $\Pi(\boldsymbol{\theta}) \subseteq S_n$ by $\Pi(\boldsymbol{\theta}) := \arg\min_{\pi \in S_n} \mathcal{L}(\boldsymbol{\theta}, \pi)$. Under Assumption 1, $\bar{\mathcal{L}}$ is locally Lipschitz and*

$$\partial \bar{\mathcal{L}}(\boldsymbol{\theta}) = \operatorname{conv}\big\{ \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \pi) \ : \ \pi \in \Pi(\boldsymbol{\theta}) \big\} \qquad \text{for every } \boldsymbol{\theta} \in \Theta.$$

In particular, the NeuroSQL algorithm satisfies:

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta\, H_t, \qquad H_t := \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^t, \pi^t) \in \partial \bar{\mathcal{L}}(\boldsymbol{\theta}^t). \tag{6}$$

The message of Lemma 2 is that the algorithm is exactly subgradient descent on $\bar{\mathcal{L}}(\boldsymbol{\theta})$, that is, on the squared Wasserstein distance between the model and the data. At any $\boldsymbol{\theta}^t$ where the optimal assignment is unique, which is the generic case, $\partial \bar{\mathcal{L}}(\boldsymbol{\theta})$ is a singleton, $\bar{\mathcal{L}}(\boldsymbol{\theta})$ is differentiable at $\boldsymbol{\theta}^t$, and the update is an ordinary gradient step.

Proposition 2 (Stationarity rate)**.** *Let Assumption 1 hold and let $\eta \le 1/\beta$. Fix the initial value $\boldsymbol{\theta}^0 \in \Theta$ and let $(H_t)_{t\ge 0}$ be any sequence satisfying:*

$$H_t \in \partial \bar{\mathcal{L}}(\boldsymbol{\theta}^t), \qquad \boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta H_t, \qquad t \ge 0. \tag{7}$$

Then, for every $T \ge 1$,

$$\frac{\eta}{2} \sum_{t=0}^{T-1} \|H_t\|^2 \ \le \ \bar{\mathcal{L}}(\boldsymbol{\theta}^0) - \bar{\mathcal{L}}(\boldsymbol{\theta}^T) \ \le \ \bar{\mathcal{L}}(\boldsymbol{\theta}^0), \tag{8}$$

and consequently:

$$\min_{0\le t<T} \|H_t\| \ \le \ \sqrt{\frac{2\,\bar{\mathcal{L}}(\boldsymbol{\theta}^0)}{\eta\, T}}. \tag{9}$$

It shows that the algorithm still attains the standard non-convex $O(1/\sqrt{T})$ stationarity rate, and that the rate does not depend on which optimal assignment the solver happens to return. This is useful in practice, since the Hungarian solver breaks ties arbitrarily. We note that this Proposition, is more general that our algorithm, since it allows H_t to be any member of the convex hull $\partial \bar{\mathcal{L}}(\boldsymbol{\theta}^t)$.

4 Distribution matching, coverage, and memorization

The experiments distinguish convergence of the empirical objective from held-out distribution quality. We evaluate fresh prior draws against independent real samples. Unless explicitly labeled, all MNIST results use the full 60,000-image training partition. More results appear in Appendix E.

Table 1: Unconditional MNIST metrics. All methods use a common generator and data split; the table reports one seed. Lower feature SW2 and Class TV are better; higher normalized entropy is better.

Method	Precision	Recall	F1	Feature SW2	Class TV	Entropy
NeuroSQL	**0.7866**	0.9038	**0.8411**	**0.0066**	**0.0380**	**0.9966**
VAE	0.6162	0.9788	0.7563	0.0158	0.1318	0.9774
GLO (shared prior)	0.4940	0.9888	0.6588	0.0175	0.1792	0.9633
GLO (fitted prior)	0.4888	0.9890	0.6542	0.0170	0.1593	0.9698

We train on all 60,000 official training images and evaluate each model using 10,000 fresh Gaussian draws per seed and a frozen classifier with 99.44% accuracy on the official test set. NeuroSQL represents all ten classes, with predicted frequencies from 8.26% to 12.00% and normalized entropy of 0.9966. GLO also represents all classes, but its class frequencies are less balanced: they range from 3.48% to 16.05% with the shared Gaussian prior, and from 4.15% to 15.12% with the fitted prior.

Figure 2 shows the predicted digit frequencies underlying the class-coverage comparison. Table 1 compares NeuroSQL and a VAE with GLO under both sampling choices. NeuroSQL has higher precision and F1, lower feature SW2, and a closer class-frequency distribution than either GLO variant. GLO has higher recall. Across three NeuroSQL seeds, F1 is 0.8458 ± 0.0077 (mean $\pm$ standard deviation, with a fixed data split).

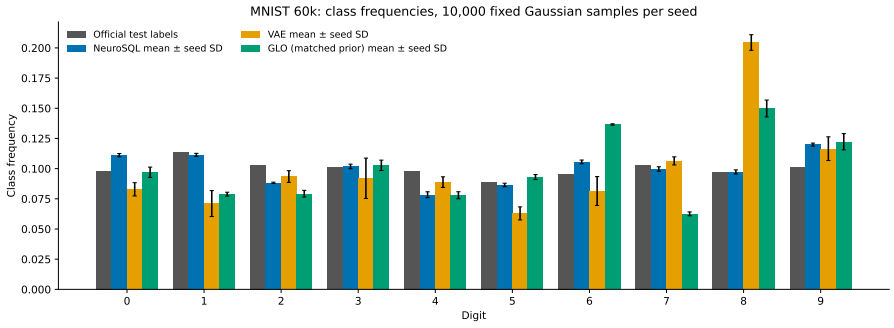

Figure 2: Predicted digit frequencies for 10,000 unconditional samples from models trained on the full 60,000-image MNIST training set. Bars show mean class frequencies across seeds for NeuroSQL, VAE, and GLO, with the official MNIST test-label distribution shown for reference.

The GLO results show high recall alongside lower precision and less balanced class frequencies. NeuroSQL's stronger F1 and lower feature SW2 indicate a different quality profile in this comparison; the table does not show that one method leads on every metric. We separately audit NEUROSQL samples for proximity to training examples. Each generated image is compared with 10,000 random images from the train set and 10,000 test images.

Table 2: Nearest-neighbor audit of 10,000 NEUROSQL-generated MNIST images. Median distances are measured to the training and test sets. "Test nearer" is the percentage of generated images closer to a test image than to a training image; the 95% intervals are Wilson intervals.

Distance space	Median train	Median test	Test nearer (%, 95% CI)	Real control (%)	Near-train (%)	Near-test (%)
Classifier features	0.0915	0.1104	14.05 [13.38, 14.74]	14.70	3.51	4.15
Pixel RMSE	0.1166	0.1368	10.59 [10.00, 11.21]	17.15	5.27	4.76

For NEUROSQL, generated samples are not systematically closer to the training set than to the test set under these measures. Near-training rates nevertheless exceed the 1% real-data control in both feature and pixel space. No exact training-image duplicates were found, either in floating-point

values or after 8-bit quantization. This audit measures sample proximity; it does not establish that memorization is absent or provide a privacy guarantee.

5 SCALABILITY

5.1 SCALING DISTRIBUTION MATCHING WITH BLOCK ASSIGNMENT

Exact global assignment is the main computational bottleneck of NEUROSQL. For n observations and n latent lattice points, it requires an $n \times n$ cost matrix and a global linear assignment solve. We therefore use block assignment, restricting each matching problem to at most B observations and latent candidates while retaining n distinct latent points overall.

Let c_G denote the cost of generating one sample and c_ℓ the cost of evaluating one candidate pair. The dominant computational terms are

$$\begin{aligned} T_{\text{global}} &= O(nc_G + n^2 c_\ell + n^3), \\ T_{\text{block}} &= O(nc_G + nBc_\ell + nB^2). \end{aligned} \tag{10}$$

Processing blocks sequentially reduces the largest stored cost matrix from $O(n^2)$ to $O(B^2)$; for fixed B, assignment work grows linearly with n. Blocking restricts the feasible coupling because observations can only be matched to candidates in their current block. Exact matching within a fixed block partition therefore minimizes a restricted assignment objective, not the global one. When partitions change, the trajectory is not equivalent to repeated global optimal assignment, so we assess scalability together with held-out distribution quality. This approximation enables full-dataset image training: on MNIST, NEUROSQL uses all $60{,}000$ training images together with $60{,}000$ distinct latent lattice points, without constructing a global $60{,}000 \times 60{,}000$ assignment problem.

5.2 MILLION-POINT SCALING ON CONTROLLED DISTRIBUTIONS

We first study scalability in controlled settings where distributional error can be measured directly. Nonlinear transport, 2D Swiss roll, and two moons are evaluated at up to $1{,}000{,}000$ observations.

Table 3 reports the million-point results. Over 30 assignment/regression updates, held-out SW2 decreases from 0.5785 to 0.0227 on Swiss roll and from 0.2352 to 0.0298 on two moons. We also evaluate a 3D Swiss roll; additional trajectories and qualitative results appear in Appendix C.1.

Figure 3 compares global Hungarian assignment at $n = 10{,}000$ with block assignment at $n \in \{100{,}000, 500{,}000, 1{,}000{,}000\}$. The larger settings retain one distinct latent point per observation while keeping the matching memory bounded by the block size.

These experiments demonstrate scalable assignment in settings with known target distributions. They do not imply recovery of the globally optimal coupling under block assignment, motivating the image-scale evaluation below.

5.3 FULL-MNIST TRAINING WITH BLOCK ASSIGNMENT

We next train NEUROSQL on the complete $60{,}000$-image MNIST training set using a fixed lattice of $60{,}000$ distinct latent points. This preserves the one-to-one latent construction while avoiding a dense global assignment problem.

Figure 4(**Top**) shows three fixed 120-epoch runs with latent dimension $d = 3$ and generator width 64. Reconstruction loss and assignment cost decrease steadily, while held-out feature-space SW2 drops rapidly and remains consistent across seeds.

The practical image implementation uses a task-adapted reconstruction loss and a separate assignment cost, so these trajectories are not instances of the exact same-cost Wasserstein descent result of Section 3. Instead, they show stable block-assignment training and decreasing held-out distributional discrepancy at full-MNIST scale. Together, the synthetic and MNIST experiments show that block assignment supports both million-observation controlled experiments and high-dimensional image training with $60{,}000$ distinct observation–latent pairs.

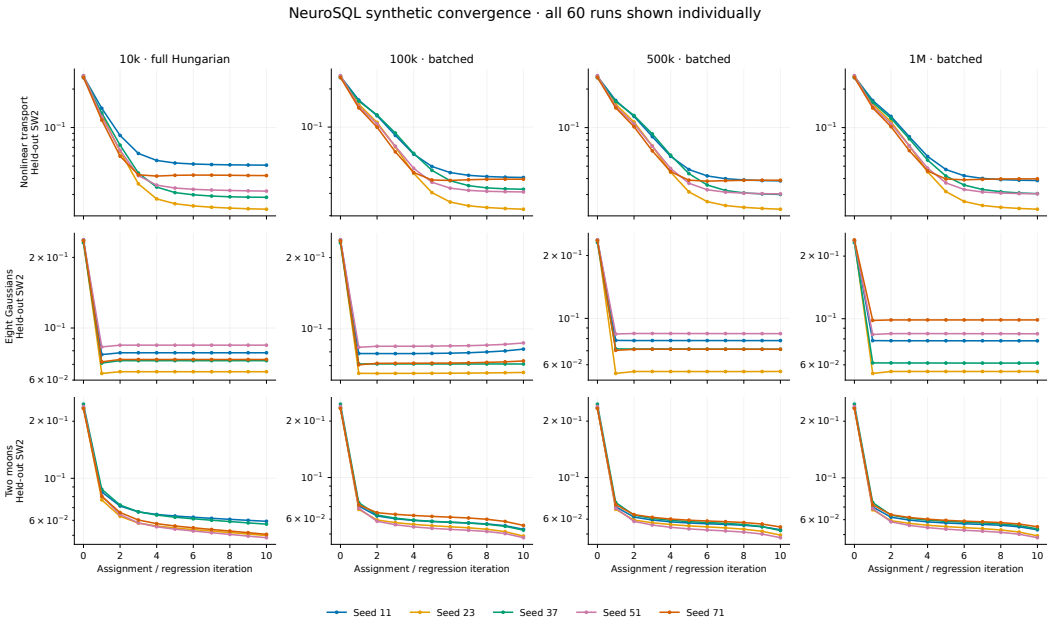

Figure 3: Held-out SW2 over assignment/regression iterations for nonlinear transport, eight Gaussians, and two moons. Columns compare 10,000 observations with full Hungarian assignment against 100,000, 500,000, and one million observations with block assignment. Each panel shows five individual seed traces.

Table 3: Million-point block-assignment results. Final SW2 is measured against independent target samples; lower is better.

Target	Final SW2	Matching time	Matrix
Nonlinear transport	0.0385	12.6 min	0.5 MiB
Swiss roll	0.0227	18.2 min	0.5 MiB
Two moons	0.0298	23.4 min	0.5 MiB

Table 4: CelebA held-out metrics, mean $\pm$ sample standard deviation over seeds 11, 23, and 47.

Method	FID↓	Prec.↑	Rec.↑	Dens.↑
GLO	52.11 ± 0.97	0.588 ± 0.008	0.054 ± 0.003	0.617 ± 0.020
VAE	57.15 ± 1.88	0.552 ± 0.006	0.055 ± 0.004	0.502 ± 0.015
NEUROSQL	$\mathbf{20.21 \pm 0.95}$	$\mathbf{0.692 \pm 0.004}$	$\mathbf{0.388 \pm 0.015}$	$\mathbf{0.890 \pm 0.007}$

Rectified Flow, MeanFlow, and FORT (Liu et al., 2023; Geng et al., 2025; Rehman et al., 2025) are also discussed more thoroughly in the appendix for context. Specialized low-data GANs and competitive one-step generators remain unevaluated image baselines.

6 Conclusion and Limitations

We introduced NEUROSQL, a generative modeling framework that replaces stochastic encoders with deterministic assignment of observations to a fixed lattice of latent quantiles; the generator itself is an ordinary regression network, and the novelty lies in the training principle. With the squared Euclidean cost, the training objective is exactly the empirical Wasserstein-2 distance between model and data, and training is subgradient descent on it: every step is non-increasing, and the iterates reach a stationary point at rate $O(T^{-1/2})$. Block assignment scales training to the full MNIST training set

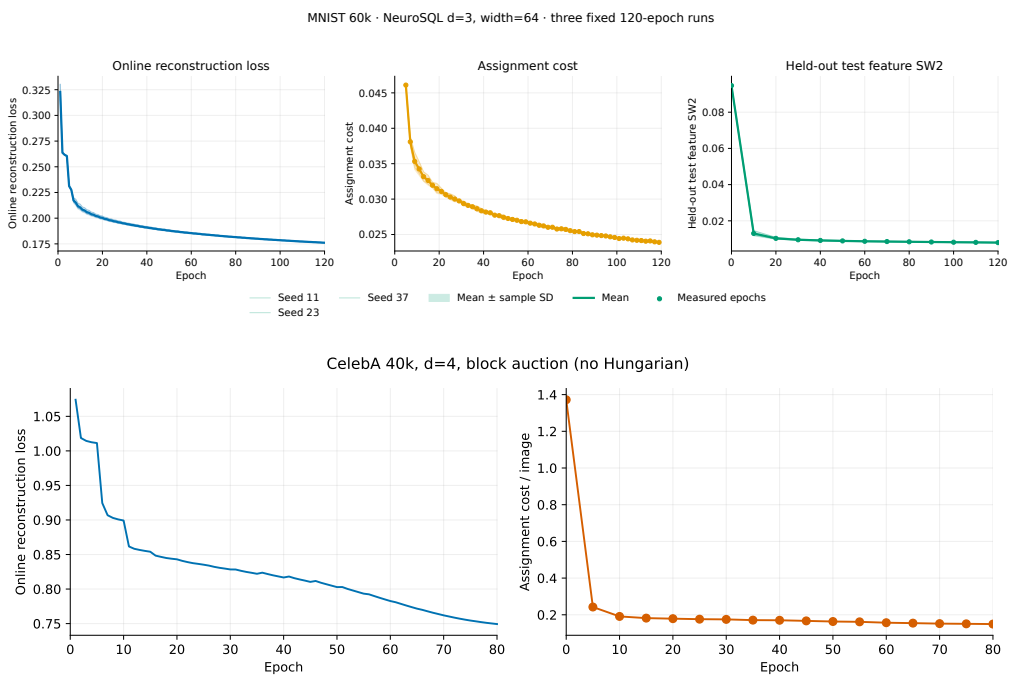

Figure 4: Training dynamics of scalable NEUROSQL on image datasets. **Top:** Full MNIST training with 60,000 images and 60,000 distinct latent points. **Bottom:** CelebA training with 40,000 images in a $d = 4$ latent space using block assignment without a global Hungarian solve. Stable optimization is shown on scaled experiments on image datasets where exact global assignment is impractical.

and to one million observations in simulations, and NEUROSQL reaches an FID of 20.21 ± 0.95 on CelebA, without relying on the costly exact Hungarian assignment.

Our guarantees are finite-sample: they establish neither global optimality, population consistency, nor improvement on unseen data, and they require a smooth generator. The one-to-one coupling prescribes the latent marginal but not coverage in output space, and training on a finite lattice while sampling from the continuous prior leaves the tails of the prior untrained at finite n. Finally, we do not characterize how far block assignment lies from exact assignment, nor how to choose the prior, lattice, or cost. These questions, and evaluation on higher-resolution benchmarks, are left for future work.

REFERENCES

Gérard Biau, Benoît Cadre, Maxime Sangnier, and Ugo Tanielian. Some theoretical properties of gans. *The Annals of Statistics*, 48(3):1539–1566, 2020.

Julien Bodelet, Guillaume Blanc, Jiajun Shan, Graciela Muniz Terrera, and Oliver Y Chén. Statistical quantile learning for large additive latent variable models. *Journal of the American Statistical Association*, 121(553):465–476, 2026. doi: 10.1080/01621459.2025.2526697.

Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 600–609. PMLR, 2018. URL https://proceedings.mlr.press/v80/bojanowski18a.html.

Minwoo Chae, Dongha Kim, Yongdai Kim, and Lizhen Lin. A likelihood approach to nonparametric estimation of a singular distribution using deep generative models. *Journal of Machine Learning Research*, 24(77):1–42, 2023.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325, 2022.

Xiaohong Chen. Large sample sieve estimation of semi-nonparametric models. In *Handbook of Econometrics*, volume 6, pp. 5549–5632. Elsevier, 2007.

Victor Chernozhukov, Alfred Galichon, Marc Hallin, and Marc Henry. Monge–kantorovich depth, quantiles, ranks and signs. *The Annals of Statistics*, 45:223–256, 2017.

Frank H. Clarke. *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, 1990. doi: 10.1137/1.9781611971309.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025. URL https://arxiv.org/abs/2505.13447.

Promit Ghosal and Bodhisattva Sen. Multivariate ranks and quantiles using optimal transport: Consistency, rates and nonparametric testing. *The Annals of Statistics*, 50(2):1012–1037, 2022.

Pegah Golestaneh, Mahsa Taheri, and Johannes Lederer. How many samples are needed to train a deep neural network? In *The Thirteenth International Conference on Learning Representations*, 2025.

Ian 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.

Marc Hallin. Measure transportation and statistical decision theory. *Annual Review of Statistics and Its Application*, 9(1):401–424, 2022.

Marc Hallin, Eustasio Del Barrio, Juan Cuesta-Albertos, and Carlos Matrán. Distribution and quantile functions, ranks and signs in dimension d: A measure transportation approach. *The Annals of Statistics*, 49:1139–1165, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6114.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

Sangyun Lee, Zinan Lin, and Giulia Fanti. Improving the training of rectified flows. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Ke Li and Jitendra Malik. Implicit maximum likelihood estimation. *arXiv preprint arXiv:1809.09087*, 2018.

Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized GAN training for high-fidelity few-shot image synthesis. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=1Fqg133qRaI.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 3481–3490, 2018.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. *Advances in Neural Information Processing Systems*, 32, 2019.

Danyal Rehman, Oscar Davis, Jiarui Lu, Jian Tang, Michael Bronstein, Yoshua Bengio, Alexander Tong, and Avishek Joey Bose. FORT: Forward-only regression training of normalizing flows. *arXiv preprint arXiv:2506.01158v1*, 2025. URL https://arxiv.org/abs/2506.01158v1.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *The Annals of Statistics*, 48(4):1875–1897, 2020.

Charles J Stone. Additive regression and other nonparametric models. *The Annals of Statistics*, 13(2): 689–705, 1985.

Taiji Suzuki. Adaptivity of deep relu network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. In *The 7th International Conference on Learning Representations (ICLR2019)*, volume 7, 2019.

Eloi Tanguy. Convergence of SGD for training neural networks with sliced wasserstein losses. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=aqqfB3p9ZA.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

AI USE STATEMENT

Generative AI tools assisted with revising portions of the manuscript, editorial analysis, manuscript restructuring, clarity revisions, and LaTeX formatting. They also provided feedback on how results and limitations were presented. Their use was not limited to sentence-level grammar correction, but also phrase suggestions. No new results were generated through AI assistance. The authors reviewed the AI-assisted material and take responsibility for the final text, numerical results, interpretations, and submission artifacts.

REPRODUCIBILITY STATEMENT

Appendix B gives the assumptions and full proofs. Appendix C specifies the experimental protocols, sample sizes, and evaluation measures. Additional matched-backbone image benchmarks appear in Appendix E.

CODE AVAILABILITY

An anonymized repository for this work is available at `https://anonymous.4open.science/r/Generative-Model-via-Quantile-Assingment/README.md`.

A MULTIVARIATE QUANTILES ($d > 1$)

We discuss how NEUROSQLapproximates the latent space in higher dimensions and show that the approximation error vanishes as $n \to \infty$. As there is no canonical ordering when $d > 1$, several methods have been developed to construct "multivariate quantiles". We concentrate on recent developments that leverage the optimal transport approach (Hallin, 2022; Chernozhukov et al., 2017; Hallin et al., 2021; Ghosal & Sen, 2022), which offer a conceptually clean and practical way to define quantiles in multivariate dimensions.

To build those multivariate quantiles, we use a regular grid, $\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_n \in \mathcal{U}_d$, where $\mathcal{U}_d := \{\boldsymbol{z} \in \mathbb{R}^d : \|\boldsymbol{z}\| < 1\}$ is the unit ball. This grid does not have to be perfectly regular, as it is in general not possible for $d \geq 3$. We require only that the discrete distribution with probability $1/n$ at each grid point converges weakly to the uniform distribution over $\mathcal{U}_d$. In practice, it is suitable to select a grid with a low discrepancy in order to obtain fast convergence rates.

In the case where the latent variables are uniformly distributed over $\mathcal{U}_d$, we define the multivariate quantiles as $\boldsymbol{Q}_i^n := \boldsymbol{U}_i$. For more general distributions P_Z, we define the multivariate quantiles as $\boldsymbol{Q}_i^n := F_\pm^{-1}(U_i)$, where $F_\pm$ denotes the center-outward distribution function. Specifically, $F_\pm$ is the unique gradient of a convex function pushing P_Z forward to the uniform distribution over the unit ball. We refer to Hallin et al. (2021) and Hallin (2022) for a detailed explanation.

The next proposition shows that the approximation error also vanishes asymptotically in the multivariate case. First, we require the following assumption.

Condition 1 (Latent distribution)**.** *The latent distribution P_Z satisfies $\mathbb{E}\|\boldsymbol{Z}\|^2 < \infty$, and its center-outward quantile map $F_\pm^{-1}$ is continuous P_U-almost everywhere.*

Condition 2 (Grid)**.** *The grid $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_n \in \mathcal{U}_d$ satisfies $\hat{P}_U^n := \frac{1}{n}\sum_{j=1}^n \delta_{\boldsymbol{U}_j} \Rightarrow P_U$ as $n \to \infty$, where P_U is the uniform distribution on $\mathcal{U}_d := \{\boldsymbol{u} \in \mathbb{R}^d : \|\boldsymbol{u}\| < 1\}$.*

Condition 3 (Second moments)**.** *The quantiles $\boldsymbol{Q}_j^n := F_\pm^{-1}(\boldsymbol{U}_j)$ satisfy $\frac{1}{n}\sum_{j=1}^n \|\boldsymbol{Q}_j^n\|^2 \to \mathbb{E}\|\boldsymbol{Z}\|^2$ as $n \to \infty$.*

Condition 1 imposes two requirements on the latent distribution. The moment condition $\mathbb{E}\|\boldsymbol{Z}\|^2 < \infty$ is needed for W_2 to be defined at all and, through the strong law of large numbers, gives convergence of the sample second moments. Continuity of $F_\pm^{-1}$ holds whenever P_Z has convex support and a density bounded away from 0 and ∞ on it, by Caffarelli regularity; Hallin et al. (2021) give conditions under which $F_\pm$ is a homeomorphism. Both are satisfied by the distributions ordinarily used as latent priors, such as the Gaussian and the uniform on a convex body.

Condition 2 is the defining requirement of the grid and is the multivariate analogue of taking the points $(j - \frac{1}{2})/n$ in dimension one. It is met by any low-discrepancy construction, and it is deliberately

weak: we do not require the grid to be regular, which is in general impossible for $d \geq 3$, only that it fills the ball in the limit.

Condition 3 controls the interaction between the two. It does *not* follow from Condition 2 when P_Z has unbounded support, because $F_{\pm}^{-1}$ diverges at the boundary of $\mathcal{U}_d$: a grid may fill the ball in the weak sense while placing enough points close to the sphere that the transported second moments diverge. The condition rules this out, and constrains the grid only near the boundary, relative to the tail of P_Z. When $P_Z = P_U$ the situation simplifies: $F_{\pm} = \mathrm{id}$, $\boldsymbol{Q}_j^n = \boldsymbol{U}_j$, continuity in Condition 1 is immediate, and Condition 3 follows from Condition 2 by boundedness of $\mathcal{U}_d$. All three conditions therefore hold automatically in the uniform case.

Proposition 3 (Asymptotic vanishing of the assignment cost)**.** *Let $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n$ be i.i.d. with distribution P_Z and assume that conditions 1, 2, and 3 hold. Then*

$$\min_{\pi \in S_n} \frac{1}{n} \sum_{i=1}^{n} \left\| \boldsymbol{Z}_i - \boldsymbol{Q}_{\pi(i)}^n \right\|^2 \longrightarrow 0 \qquad \textit{almost surely as } n \to \infty.$$

B Proofs

B.1 Proof of Lemma 1.

Proof. Both $\hat{\mu}_n$ and $G_{\theta\#}\hat{\nu}_n$ are uniform measures on n atoms, so a coupling of the two is an $n \times n$ matrix P with $P_{ij} \geq 0$ and row and column sums equal to $1/n$; that is, $P \in \frac{1}{n}\mathcal{B}_n$, where $\mathcal{B}_n$ denotes the Birkhoff polytope of doubly stochastic matrices. Writing $C_{ij} = \|X_i - G_\theta(Q_j)\|^2$, the Kantorovich problem is:

$$W_2^2\big(\hat{\mu}_n, G_{\theta\#}\hat{\nu}_n\big) = \min_{P \in \frac{1}{n}\mathcal{B}_n} \langle P, C \rangle.$$

Since the objective is linear in P and $\mathcal{B}_n$ is compact and convex, the minimum is attained at a vertex of $\mathcal{B}_n$. By the Birkhoff–von Neumann theorem the vertices of $\mathcal{B}_n$ are exactly the $n!$ permutation matrices $\{P_\pi : \pi \in S_n\}$, where $(P_\pi)_{ij} = \mathbf{1}\{j = \pi(i)\}$. Since $\langle \frac{1}{n} P_\pi, C \rangle = \frac{1}{n} \sum_i \|X_i - G_\theta(Q_{\pi(i)})\|^2 = L(\theta, \pi)$, minimising over couplings and over S_n yield the same value, i.e. $\min_{\pi \in S_n} L(\theta, \pi) = W_2^2(\hat{\mu}_n, G_{\theta\#}\hat{\nu}_n)$. □

B.2 Proof of Proposition 1.

Proof. By the Hungarian algorithm, we have:

$$\mathcal{F}(\theta^{t+1}, \pi^{t+1}) \;\leq\; \mathcal{F}(\theta^{t+1}, \pi),$$

for all π, in particular π^t. Then, given π^t, the gradient Lemma gives:

$$\mathcal{F}(\theta^{t+1}, \pi^t) \leq \mathcal{F}(\theta^t, \pi^t) - \eta(1 - \beta\eta/2)\|\nabla_\theta \mathcal{F}(\theta^t, \pi^t)\|.$$

The Proposition then holds for $0 < \eta < 1/\beta$. □

B.3 Proof of Lemma 2.

Proof. Each $\mathcal{F}(\cdot, \pi)$ is continuously differentiable by Assumption 1 and S_n is finite, so $\bar{\mathcal{F}} = \min_{\pi \in S_n} \mathcal{F}(\cdot, \pi)$ is a minimum of finitely many locally Lipschitz functions and is therefore locally Lipschitz; Definition 1 applies.

Write $-\bar{\mathcal{F}} = \max_{\pi \in S_n} -\mathcal{F}(\cdot, \pi)$ and note that the maximum is attained exactly on $\Pi(\theta)$. Clarke's rule for a finite maximum of continuously differentiable functions (Clarke, 1990) gives:

$$\partial\big(-\bar{\mathcal{F}}\big)(\theta) = \mathrm{conv}\,\big\{ -\nabla_\theta \mathcal{F}(\theta, \pi) : \pi \in \Pi(\theta) \big\},$$

where conv denotes the convex hull. The next part of the Lemma follows by noting that $\partial(-\mathcal{F}) = -\partial\mathcal{F}$ and $\mathrm{conv}(-A) = -\,\mathrm{conv}(A)$.

Finally, the iterates of our Algorithm satisfy $\pi^t \in \Pi(\theta^t)$ by definition. $H^t = \nabla_\theta \mathcal{F}(\theta^t, \pi^t)$ is one of the vectors generating the convex hull above. □

B.4 Proof of Proposition 2.

Proof. The argument has two steps: a one-step decrease valid for an arbitrary admissible selection, then telescoping.

Step 1: step decrease. Let $t \geq 0$, $H_t \in \partial\bar{\mathcal{F}}(\theta^t)$, and $m = |\partial\bar{\mathcal{F}}(\theta)|$. By Lemma 2 there exist $\pi_1, \ldots, \pi_m \in \Pi(\theta^t)$ and weights $\alpha_1, \ldots, \alpha_m \geq 0$ with $\sum_{k=1}^m \alpha_k = 1$ such that:

$$H_t = \sum_{k=1}^{m} \alpha_k \, \nabla_\theta \mathcal{F}(\theta^t, \pi_k). \tag{11}$$

Each $\mathcal{F}(\cdot, \pi_k)$ has β-Lipschitz gradient, so the descent Lemma applied at θ^t along the increment $\theta^{t+1} - \theta^t = -\eta H_t$ gives, for every k,

$$\mathcal{F}(\theta^{t+1}, \pi_k) \leq \mathcal{F}(\theta^t, \pi_k) - \eta\big\langle \nabla_\theta \mathcal{F}(\theta^t, \pi_k), H_t \big\rangle + \frac{\beta\eta^2}{2}\|H_t\|^2.$$

Multiplying the k-th inequality by α_k and summing over k, we use two facts. First, every π_k belongs to $\Pi(\theta^t)$, so $\mathcal{F}(\theta^t, \pi_k) = \bar{\mathcal{F}}(\theta^t)$ for each k and hence $\sum_k \alpha_k \mathcal{F}(\theta^t, \pi_k) = \bar{\mathcal{F}}(\theta^t)$. Second, by linearity of the inner product and equation 11,

$$\sum_k \alpha_k \big\langle \nabla_\theta \mathcal{F}(\theta^t, \pi_k), H_t \big\rangle = \Big\langle \sum_k \alpha_k \nabla_\theta \mathcal{F}(\theta^t, \pi_k), H_t \Big\rangle = \|H_t\|^2.$$

Therefore:

$$\sum_k \alpha_k \, \mathcal{F}(\theta^{t+1}, \pi_k) \leq \bar{\mathcal{F}}(\theta^t) - \eta\Big(1 - \frac{\beta\eta}{2}\Big)\|H_t\|^2 \leq \bar{\mathcal{F}}(\theta^t) - \frac{\eta}{2}\|H_t\|^2,$$

with the last inequality using $\eta \leq 1/\beta$. Since $\bar{\mathcal{F}}$ is a minimum over all of S_n, we have $\bar{\mathcal{F}}(\theta^{t+1}) \leq \mathcal{F}(\theta^{t+1}, \pi_k)$ for every k; taking the convex combination of these inequalities with weights α_k gives $\bar{\mathcal{F}}(\theta^{t+1}) \leq \sum_k \alpha_k \mathcal{F}(\theta^{t+1}, \pi_k)$. Combining,

$$\bar{\mathcal{F}}(\theta^{t+1}) \;\leq\; \bar{\mathcal{F}}(\theta^t) - \frac{\eta}{2}\|H_t\|^2. \tag{12}$$

Step 2: Telescoping. Summing equation 12 over $t = 0, \ldots, T-1$ and rearranging gives:

$$\frac{\eta}{2}\sum_{t=0}^{T-1} \|H_t\|^2 \;\leq\; \bar{\mathcal{F}}(\theta^0) - \bar{\mathcal{F}}(\theta^T) \;\leq\; \bar{\mathcal{F}}(\theta^0),$$

with the last inequality due to $\bar{\mathcal{F}} \geq 0$. This is equation 8. Bounding the minimum by the average, we have:

$$T \min_{0 \leq t < T} \|H_t\|^2 \leq \sum_{t=0}^{T-1} \|H_t\|^2 \leq \frac{2\,\bar{\mathcal{F}}(\theta^0)}{\eta},$$

and, by taking square roots, we have equation 9. We note that as no constant in the argument depends on the particular selection (H_t), the bound holds uniformly over admissible sequences. □

B.5 Proof of Proposition 3

Proof. Write $\tilde{P}_Z^n := \frac{1}{n}\sum_{i=1}^n \delta_{\boldsymbol{Z}_i}$ for the empirical measure of the sample and $\hat{P}_Z^n := \frac{1}{n}\sum_{j=1}^n \delta_{\boldsymbol{Q}_j^n}$ for the measure carried by the quantile grid.

From the arguments of Lemma 1, we have

$$\min_{\pi \in S_n} \; \frac{1}{n}\sum_{i=1}^{n} \big\|\boldsymbol{Z}_i - \boldsymbol{Q}^n_{\pi(i)}\big\|^2 \;=\; W_2^2\big(\tilde{P}_Z^n,\, \hat{P}_Z^n\big). \tag{13}$$

By $\hat{P}_Z^n = (F_\pm^{-1})_\# \hat{P}_U^n$, Condition 2, 2, and the continuous mapping theorem,

$$\hat{P}_Z^n \;\Longrightarrow\; (F_\pm^{-1})_\# P_U \;=\; P_Z.$$

Table 5: Final held-out SW2 after 30 iterations, seed 11. Training size and assignment regime change together.

Target	10k global	100k block	500k block	1M block	Real–real
Nonlinear transport	.0505	.0399	.0383	.0385	.0230
Swiss roll (2D)	.0269	.0236	.0226	.0227	.0225
Two moons	.0514	.0281	.0287	.0298	.0131
Swiss roll (3D)	.0241	.0266	.0236	.0243	.0192

Together with the convergence of second moments in Condition 2, weak convergence upgrades to convergence in W_2 (Villani et al., 2009, Thm. 6.9), so $W_2(\hat{P}_Z^n, P_Z) \to 0$.

By Varadarajan's theorem, $\tilde{P}_Z^n \Rightarrow P_Z$ almost surely. Since $\mathbb{E}\|\boldsymbol{Z}\|^2 < \infty$, the strong law of large numbers gives $\frac{1}{n}\sum_{i=1}^n \|\boldsymbol{Z}_i\|^2 \to \mathbb{E}\|\boldsymbol{Z}\|^2$ almost surely. Applying (Villani et al., 2009, Thm. 6.9) again, $W_2(\tilde{P}_Z^n, P_Z) \to 0$ almost surely.

By the triangle inequality for the metric W_2,

$$W_2\big(\tilde{P}_Z^n, \hat{P}_Z^n\big) \;\le\; W_2\big(\tilde{P}_Z^n, P_Z\big) + W_2\big(P_Z, \hat{P}_Z^n\big) \;\longrightarrow\; 0$$

almost surely. The claim follows from equation 13. □

C Experimental protocols and additional diagnostics

This appendix specifies the experimental protocols used for the synthetic distribution-matching and MNIST studies, together with additional diagnostics for optimization, distribution matching, and sample proximity.

C.1 Synthetic distribution-matching study

The synthetic study considers nonlinear Gaussian transport, two noisy moons, a noisy interleaved 2D Swiss roll, and a 3D Swiss roll. For each target, $n \in \{10{,}000, 100{,}000, 500{,}000, 1{,}000{,}000\}$ observations are fitted for 30 assignment/regression iterations. A scrambled Sobol lattice is mapped through the standard-normal inverse CDF to construct the latent points. The $n = 10{,}000$ setting uses global Hungarian assignment, whereas the larger settings use spatial Hungarian blocks of at most 256 points. Every observation and every distinct lattice atom participates in each iteration; blocking restricts candidate pairs while preserving a permutation of the full lattice.

The decoders are target-specific low-dimensional basis models: a nonlinear basis for the transport target, two routed trigonometric experts with prescribed two-arc geometry for the moons, and spiral/manifold bases for the 2D and 3D Swiss rolls. Matched-pair regression uses ridge coefficient 10^{-8}. Simulator latents are used only to generate training and evaluation observations and do not enter either assignment or fitting.

For block assignment, spatially localized candidate sets are formed before solving the within-block Hungarian problems. This restricts the feasible coupling relative to global assignment but keeps the largest cost matrix bounded by the block size. The 10k global-Hungarian setting provides an exact-assignment reference at a scale where the full cost matrix remains tractable.

Held-out sliced Wasserstein-2 (SW2) is evaluated using 8,192 independently generated observations and 8,192 independent target observations with 64 fixed projection directions. A separate independent target draw provides the real–real finite-sample reference. Evaluation prior codes are held fixed throughout each run.

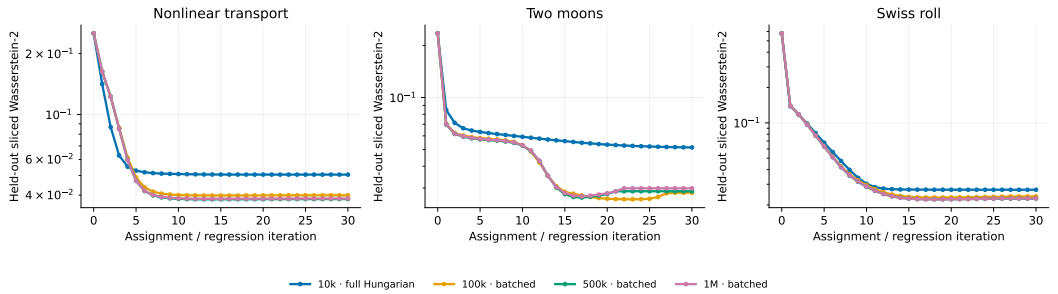

Figure 5: Held-out SW2 over 30 assignment/regression iterations for the synthetic scaling experiments. The 10k setting uses global Hungarian assignment, while the 100k, 500k, and 1M settings use spatial block assignment.

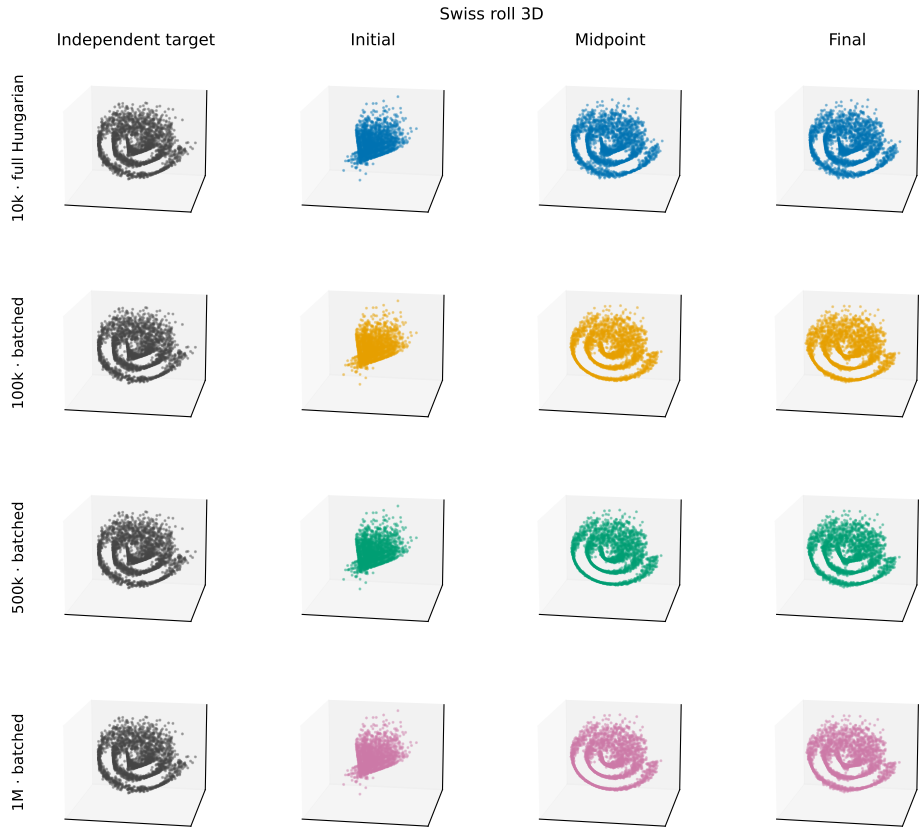

Figure 6: Qualitative evolution on the 3D Swiss roll across $n \in \{10\text{k}, 100\text{k}, 500\text{k}, 1\text{M}\}$. Columns show the generated point clouds at the initial, midpoint, and final assignment/regression iterations.

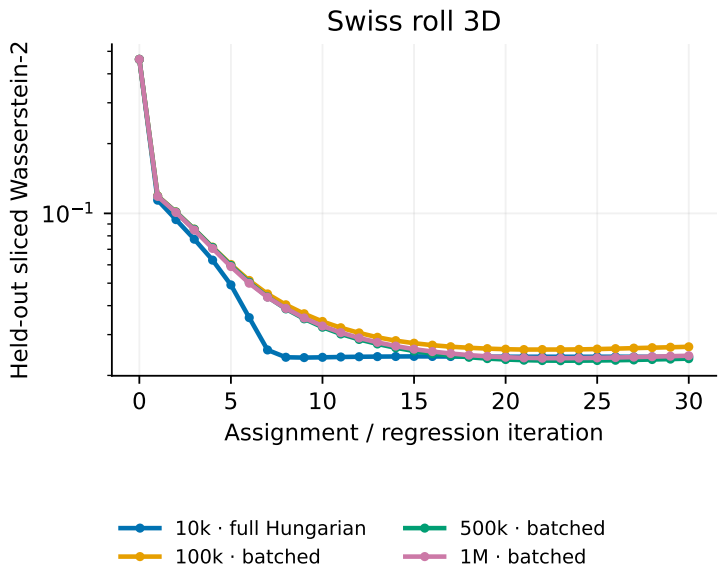

Figure 7: Training progression on the 3D Swiss roll across $n \in \{10\text{k}, 100\text{k}, 500\text{k}, 1\text{M}\}$. Global Hungarian assignment is used at 10k and spatial block assignment at the larger sample sizes.

C.2 Full-MNIST training and evaluation

All MNIST experiments use the complete official training partition of $60{,}000$ images for generator fitting. The official $10{,}000$-image test partition is reserved for evaluation. The reported NeuroSQL experiments use a three-dimensional Gaussian latent space and a width-64 generator. We train three fixed runs with seeds 11, 23, and 37 for 120 epochs using identical hyperparameters.

NeuroSQL uses a fixed scrambled Sobol–Gaussian lattice containing $60{,}000$ distinct latent atoms, one for each training observation. Assignment is performed with block matching rather than a global $60{,}000 \times 60{,}000$ Hungarian solve. The assignment procedure retains a one-to-one permutation of the full $60{,}000$-point lattice while restricting the candidate pairs considered within each block. Consequently, the number of represented latent locations grows with the full training set rather than with the block size.

For image training, the generator is optimized with the task-specific reconstruction objective, while the assignment step uses its corresponding image-matching cost.

Final unconditional evaluation uses $10{,}000$ fixed i.i.d. Gaussian latent draws per seed, independently of the $60{,}000$ lattice points used during training. The same evaluation protocol is applied to NeuroSQL, VAE, and GLO. Class frequencies are obtained from predictions of the frozen MNIST classifier and are compared with the class frequencies of the official test set. Reported uncertainty across the three runs is the sample standard deviation over seeds 11, 23, and 37.

Figure 4 separates the three quantities used to assess the practical training dynamics. Reconstruction loss and assignment cost decrease steadily over training. Held-out feature SW2, computed from fresh prior samples against official test images, decreases rapidly during the initial epochs and then improves more gradually. The three training runs follow closely aligned trajectories throughout the 120 epochs.

C.3 Projected feature-distribution diagnostic

To complement the scalar distributional metrics, we compare the generated and real feature distributions along the leading principal-component direction of the frozen classifier representation. For each of the three NeuroSQL seeds, the marginal density obtained from $10{,}000$ fresh Gaussian samples is compared with the corresponding density of the official MNIST test set. The same feature projection and plotting procedure are used for all three runs.

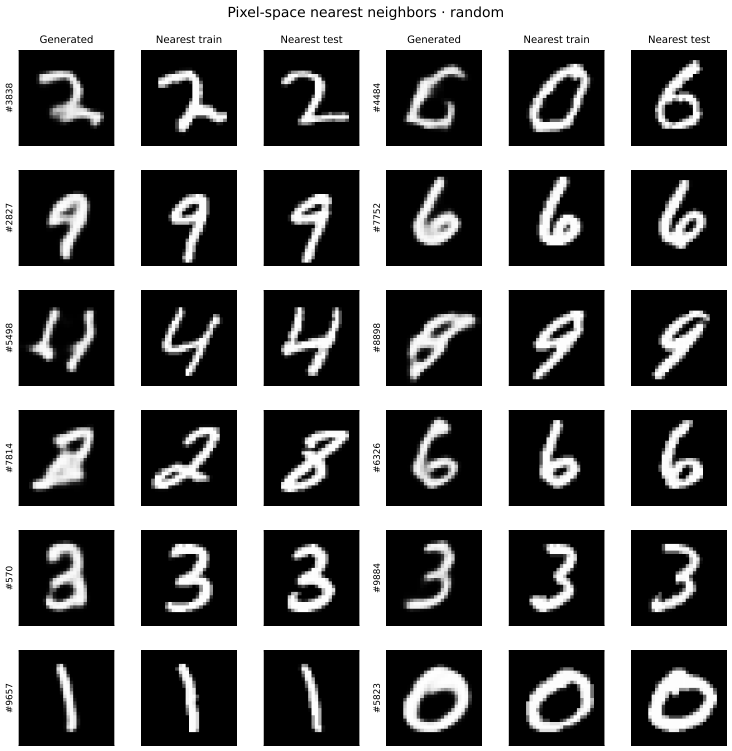

Figure 8: Pixel-space nearest-neighbor audit for random NeuroSQL samples on MNIST. Each generated image is shown alongside its nearest neighbor from the full 60,000-image training set and from the official 10,000-image test set, measured by pixel-space Euclidean distance. The examples illustrate that generated samples are generally similar to real digits without being visually identical to their closest training examples.

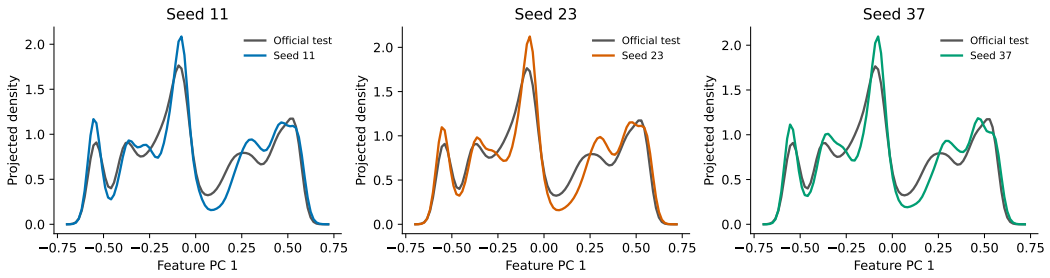

Figure 9: Projected feature-density comparison for NeuroSQL trained on all 60,000 MNIST training images. Each panel compares the density of 10,000 fresh unconditional samples from one training seed with the official MNIST test distribution along the first principal-component direction of the frozen classifier features.

The projected marginals reproduce the principal modes of the real feature distribution across all three training seeds. This visualization is a one-dimensional diagnostic of distributional alignment and is complementary to, rather than a substitute for, the feature-space SW2 and class-coverage metrics.

Algorithm 2 Selected MNIST NeuroSQL training procedure

Input: training images $X_{1:n}$, fixed lattice $Q_{1:n}$, block cap $B = 512$, 80 epochs
Fit label-free training-descriptor PCA; initialize permutation π from PCA-whitened scores and $Q_{1:n}$
Initialize generator parameters θ and EMA copy $\bar{\theta}$
for epoch $e = 1, \ldots, 80$ **do**
    if $e \geq 5$ and $e - 5$ is divisible by 2 **then**
        Generate $G_\theta(Q_{1:n})$ and project generated descriptors with the fixed PCA
        Form a spatial block-Hungarian proposal; accept it if its descriptor cost improves on π
        Refine π by exact Hungarian solves within random blocks of currently owned lattice indices
    end if
    for each minibatch of 128 observations **do**
        Set $z_i = Q_{\pi(i)} + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, .01^2 I)$
        Take an Adam step on the compound reconstruction loss; update EMA parameters $\bar{\theta}$
    end for
end for
Output: EMA generator $G_{\bar{\theta}}$ for fresh-prior evaluation

D Extended methodological material

The main text separates the exact same-cost permutation analysis from the practical implementations. The specifics regarding the runs and diagnostics for the completed synthetic and MNIST runs are in Appendix C.

D.1 Practical assignment and generator updates

Algorithm 2 describes the selected unconditional MNIST run, that is detailed in Appendix C.2. The synthetic scaling study instead alternates assignment with fitting a target-specific basis decoder; it does not use this CNN optimizer. The algorithm here is a reproducible description of an image implementation, not the constant-step, same-cost iteration analyzed in the theory section.

Let $Q_1, \ldots, Q_n$ be the fixed Sobol–Gaussian lattice. A label-free PCA of multiscale training-image descriptors initializes a permutation. Assignment compares PCA-projected descriptors of real and generated images. At a scheduled reassignment, a spatial block-Hungarian proposal is accepted only when its descriptor cost improves on the current permutation, after which randomly grouped, ownership-preserving Hungarian blocks refine the pairing. Each step is a bijection over all n lattice indices, although neither block solve is the global optimum over all $n!$ permutations. The generator is then trained with jittered assigned codes and the compound reconstruction loss; evaluation uses EMA weights and fresh Gaussian codes.

The assignment cost is squared distance between PCA-projected multiscale descriptors; the generator loss combines SSIM, L1, BCE, and edge L1. Because the two steps optimize different costs and generator updates use Adam, minibatches, jitter, and EMA, the exact same-cost Wasserstein and constant-step descent statements do not apply to this run. EMA changes the evaluated generator, while the permutation remains defined on the fixed unsmoothed lattice. The practical run does not use the latent momentum update shown as an optional schematic operation in Figure 1.

D.2 Cost of global and block assignment

Let c_G be the cost of generating one sample and c_ℓ the cost of evaluating one candidate-pair descriptor distance. Building a dense $n \times n$ cost matrix and solving global Hungarian assignment costs

$$O(nc_G + n^2 c_\ell + n^3)$$

time and $O(n^2)$ cost-matrix memory. The pairwise construction depends on the observation or descriptor dimension through c_ℓ; assignment is therefore not end-to-end independent of image dimension.

With blocks of size at most B, one assignment pass has matching and matrix-construction terms $O(nB^2 + nBc_\ell)$, in addition to $O(nc_G)$ for generating the candidate outputs. Processing blocks sequentially reduces the largest stored cost matrix from $O(n^2)$ to $O(B^2)$. For fixed B, the assignment

work therefore grows linearly with n, although generating candidates, constructing blocks, and training the generator still process all n observations.

This reduction is what makes quantile assignment practical beyond the regime in which global Hungarian matching is feasible. In the controlled synthetic experiments, the 10k setting uses global Hungarian assignment, whereas the 100k–1M settings use blocks of at most 256 points. In the MNIST experiments, NeuroSQL is trained on the complete $60{,}000$-image training partition using block assignment with a lattice of $60{,}000$ distinct latent atoms. Thus, blocking reduces the size of the assignment problems without reducing the number of latent locations represented by the model: every training observation retains a distinct lattice point, while a dense $60{,}000 \times 60{,}000$ cost matrix and global Hungarian solve are avoided.

A block assignment is exact only within its candidate blocks. For a fixed partition, it minimizes a restricted assignment cost whose value may exceed the global minimum. When block membership changes between updates, the resulting sequence is not equivalent to repeated global optimal assignment. In the MNIST implementation, the block-assignment procedure nevertheless maintains a bijection over the full $60{,}000$-point lattice.

E ADDITIONAL IMAGE DIAGNOSTICS

This section provides additional diagnostics for the image-generation experiments reported in the main text. The MNIST analyses correspond to models trained on the complete 60,000-image official training partition. Unconditional evaluation uses 10,000 fresh Gaussian latent draws per seed, independently of the fixed 60,000-point latent lattice used during training.

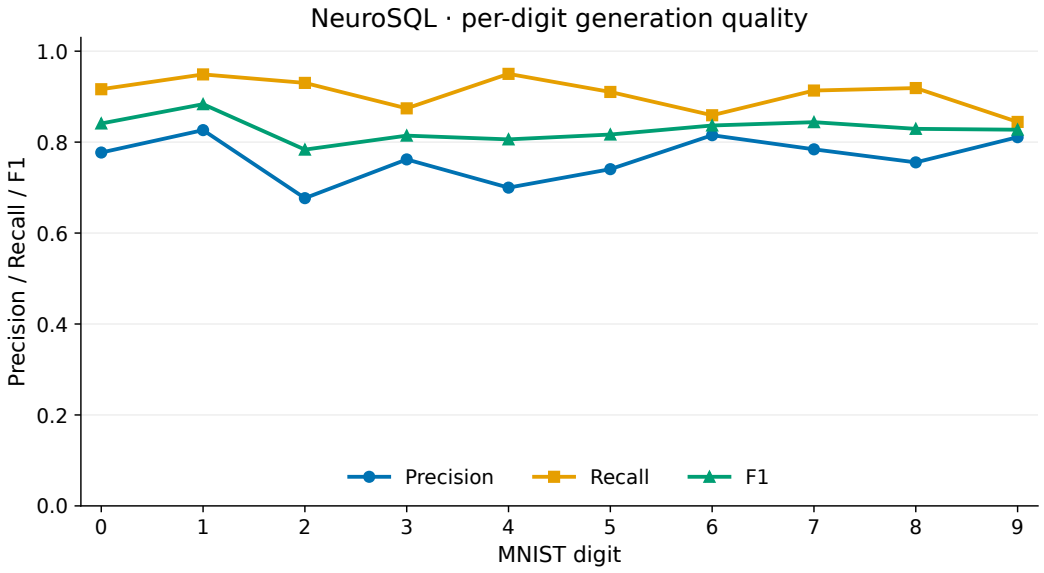

Figure 10: Class-conditional generative precision, recall, and F1 for NeuroSQL trained on all 60,000 MNIST training images. The 10,000 official test images are grouped by their true labels, while 10,000 fresh unconditional samples are grouped by frozen-classifier predictions. These quantities measure class-conditional generative fidelity and coverage and are not classifier confusion-matrix scores.

The class-conditional results complement the aggregate precision, recall, F1, class-frequency, and feature-SW2 measures reported in the main text. Evaluating the classes separately makes it possible to identify differences in fidelity and coverage that are not visible from global averages alone.

CelebA evaluation. The CelebA experiments are evaluated separately from MNIST using the image-generation protocol described in the main text. Table 4 reports held-out FID, precision, recall, and density over seeds 11, 23, and 47 for the matched comparison with GLO and the VAE. The larger 40k-image CelebA experiment evaluates block assignment at increased training-set scale. These results assess image-generation quality and scalability under their respective CelebA protocols and are not combined with the MNIST metrics.

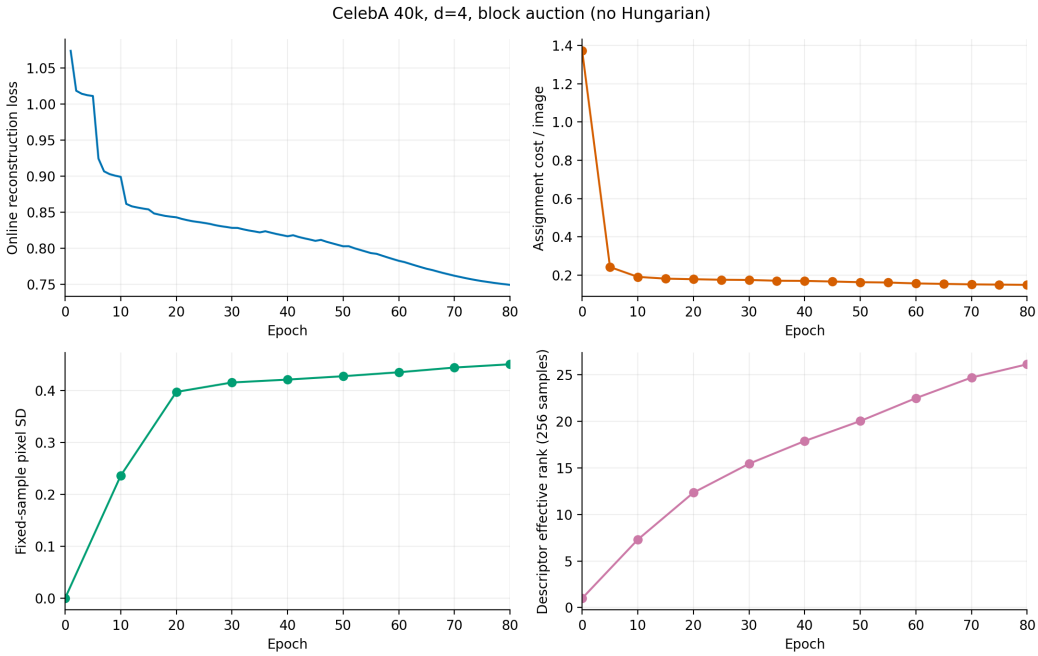

Figure 11: Training dynamics on CelebA with 40,000 training images. NeuroSQL is trained with block assignment in a $d = 4$ latent space, avoiding a global Hungarian solve. The panels show the online reconstruction loss, assignment cost per image, fixed-sample pixel standard deviation, and effective descriptor rank over 80 epochs. Reconstruction and assignment costs decrease throughout training, while the generated samples maintain non-degenerate pixel variability and increasing descriptor rank, indicating stable optimization and growing output diversity at this scale.



Figure 12: Unconditional CelebA samples after 80 epochs of training on 40,000 images. Samples are generated from fresh latent draws after training NeuroSQL with block assignment in a $d = 4$ latent space. The model produces diverse faces spanning variation in identity, pose, hairstyle, age, skin tone, and background appearance, demonstrating that the scalable block-assignment procedure remains trainable and produces varied outputs beyond the regime where global Hungarian assignment is practical.

F EXTENDED RELATED WORK

Discrete latent representations and vector quantization. Discrete latent space models, including VQ-VAE (van den Oord et al., 2017), VQ-VAE-2 (Razavi et al., 2019), and VQ-GAN (Esser et al., 2021), utilize learned codebooks to map inputs to discrete vectors, a paradigm further advanced by masked (Chang et al., 2022) and autoregressive (Tian et al., 2024) modeling. While NEUROSQL also employs discrete representations, it fundamentally departs from this framework by replacing the learned encoder and its associated sample complexity with a fixed *a priori* quantile lattice, determining embeddings via combinatorial assignment rather than gradient-based inference.

Optimal Transport and Rectified Flows. A related and rapidly evolving line of work addresses this bottleneck by combining generative modeling with optimal transport (OT). Rectified Flows (Liu et al., 2023) formulate generative modeling as learning straight ODE trajectories between noise and data distributions, with subsequent work improving training efficiency (Lee et al., 2024), achieving one-step generation (Geng et al., 2025), and enabling forward-only regression training (Rehman et al., 2025). These methods share a common paradigm: parameterizing continuous neural transport maps optimized to minimize Wasserstein distance, enabling few-step sampling. NEUROSQLuses OT fundamentally differently. Rather than learning continuous trajectories, we use OT-based multivariate quantiles (Hallin et al., 2021; Chernozhukov et al., 2017; Hallin, 2022; Ghosal & Sen, 2022) to define a fixed lattice that partitions the latent space, then solve discrete assignment problems to match data to quantiles.

DGM for low sample regimes. Existing approaches for low-sample generation, ranging from data augmentation to specialized architectures like FastGAN (Liu et al., 2021), generally remain within paradigms that require auxiliary networks to process high-dimensional data. Consequently, these methods face severe sample complexity challenges when the sample size is small relative to the data dimension ($n \ll p$). NEUROSQL addresses this bottleneck by eliminating encoders and discriminators, thereby sidestepping the difficulty of learning functions from high-dimensional observation spaces.