

A Architectures and Training Dynamics

In this section we outline some general details about the architectures and training dynamics for GDEs. In the following section we will give more detailed explanations about each specific experiment, in addition to full details available in the codebase. All of these findings are somewhat provisional, and there is significant scope for future work to further explore these design choices, but we hope this is a useful complement to our codebase for researchers trying to train their own GDEs.

A.1 Encoder Architectures

Our framework utilizes permutation-invariant encoders to map input sets $S_m = \{x_1, \dots, x_m\}$, where each $x_i \in \mathbb{R}^d$, to a fixed-dimensional latent representation $z \in \mathbb{R}^l$. We primarily employ several types of set encoders, including variants based on self-attention, Graph Neural Network (GNN)-style pooling, and residual connections. All encoders typically conclude by applying a final pooling operation (e.g., mean pooling) across the element representations, followed by a linear projection and a non-linearity (e.g., SELU) to produce the final latent vector z .

A.1.1 Simple Self-Attention Encoder

This encoder provides a baseline transformer-based approach. It first applies a linear layer followed by a SELU activation to project input elements x_i into a hidden dimension H . It then processes these representations through a series of multi-head Self-Attention blocks [57]. This architecture directly models pairwise interactions within the set.

A.1.2 Simple GNN Encoder

The simple GNN-style encoder offers an alternative based on iterative pooling and non-linear transformations, distinct from the standard DeepSets [21] sum-decomposition. It starts with an MLP projection into the hidden dimension H . Subsequently, it applies a sequence of layers, each performing a pooling operation across the set followed by an MLP. This structure iteratively refines element representations based on aggregated set information.

Pooling Operations: Our theoretical framework (see Appendix D.2) justifies the use of pooling operations that correspond to M/Z-estimators. We focus on mean pooling but additionally implement median pooling as an illustrative example. Notably, max pooling is generally not suitable in this context as its non-differentiability breaks the convergence guarantees we are interested in for Eq. (1), see the remarks in App D.2 for details. Future work might thoroughly explore which pooling operations lead to the greatest flexibility and stability for distribution embedding.

A.1.3 ResNet-GNN Encoder

To improve gradient flow and enable deeper architectures, we enhance the GNN-style encoder with residual connections. This encoder first projects each input element x_i into H using an MLP. It then processes the set through a series of blocks where each block k computes an intermediate representation $h_i^{(k)}$ for each element i . The core operation within a block uses mean pooling (or median pooling). Inspired by ResNet [58, 23], we incorporate skip connections. The input to block k includes the output from the previous block $h^{(k-1)}$, a linear projection of the original input x , and the output of the initial MLP projection. Formally:

$$h^{(k)} = \text{LayerNorm}(\text{PooledFC}(h^{(k-1)}) + h^{(k-1)} + \text{Linear}_k(x))$$

where $h^{(0)}$ is the output of the initial input projection combined with a projection of x , followed by Layer Normalization. This structure ensures the original input signal is preserved.

A.1.4 ResNet-Transformer Encoder

This variant follows the same residual structure as the ResNet-MLP encoder but replaces the layers with standard multi-head Self-Attention blocks [57]. This potentially allows the model to learn more complex interactions while benefiting from the improved training dynamics of residual connections. The skip connection mechanism remains identical to the ResNet-MLP version.

A.1.5 Encoder Comparison

Transformer-based encoders (Simple Self-Attention and ResNet-Transformer) often leverage pre-trained weights effectively and can converge in fewer epochs compared to GNN-style approaches. However, this typically comes at a higher computational cost per epoch and during inference due to the quadratic complexity of self-attention with respect to set size m . With sufficient training, we find that the GNN-based architectures, particularly the ResNet-GNN, achieve strong performance, often rivaling the transformer variants while being more computationally efficient for large sets.

Alternative Generative Strategies and Sampling The Wasserstein Wormhole [15] uses a self-attention decoder with fixed positional embeddings that can map the latent z back to samples. One potential method replaces fixed positional embeddings with samples drawn from a simple distribution (e.g., Gaussian) transforming this into a true generator. But this incurs substantial computational costs (e.g., quadratic cost in the number of generated samples for attention-based sampling decoders), and it is not clear this would lead to significant improvements in performance.

It also becomes less obvious how to adapt existing generator architectures using this approach. One option is to use self-attention to construct sample-specific conditional signals from the latent z and the noise vector, and then condition the generator on this signal. This is significantly more complex, and is not clear that this would lead to significant improvements in performance.

A.2 Adapting Pre-trained Models

Our framework is designed to flexibly incorporate pre-trained models, leveraging their learned representations and generative capabilities. We adapt pre-trained models for both the encoder and the generator components.

A.2.1 Encoder Adaptation

For tasks involving complex input modalities like natural language or protein sequences, we can utilize pre-trained transformer-based encoders such as BERT [59] or ESM [60] as powerful feature extractors. These pre-trained models can serve as the initial feature extraction layer, whose outputs $\{h_1, \dots, h_N\}$ are then fed into the subsequent aggregation layers of our set encoders (e.g., ResNet-GNN or ResNet-Transformer, see subsection A.1).

The adaptation process typically involves:

1. **Loading Pre-trained Weights:** We load the desired pre-trained encoder model using standard libraries like Hugging Faces transformers [61].
2. **Feature Extraction:** For each element x_i in the input set $X = \{x_1, \dots, x_N\}$, we pass it through the pre-trained transformer to obtain a contextualized representation h_i . Often, the output embedding corresponding to a special token (like [CLS] in BERT) or the mean/max-pooled output of the final hidden states is used.
3. **Set Aggregation:** These element-wise feature vectors $\{h_1, \dots, h_N\}$ are then fed into the subsequent layers of our chosen set encoder (e.g., ResNet-MLP or ResNet-Transformer layers) which perform the permutation-invariant aggregation to produce the final latent representation z .
4. **Fine-tuning (Optional):** Depending on the task and dataset size, the pre-trained encoder’s weights might be kept frozen initially or fine-tuned jointly with the rest of the model during end-to-end training.

A.2.2 Generator Adaptation and Conditioning

A core strength of our approach is the ability to use large pre-trained causal language models (LMs), such as GPT-2 [62], ProGen2 [29], or specialized models like HyenaDNA [30], as the conditional generator $p_\theta(x|z)$.

The adaptation involves:

1. **Loading Pre-trained Weights:** We load the chosen pre-trained causal LM and its associated tokenizer using ‘transformers’ [61].

2. **Prefix Conditioning:** The primary challenge is to effectively condition the generator's output on the latent set representation z produced by the encoder. In practice, we find prefix tuning to be an effective and widely applicable method. The latent vector $z \in \mathbb{R}^L$ is projected, typically via a small MLP W_p , into one or more vectors $p = W_p(z)$ that have the same hidden dimension as the LM. These projected vectors p are then treated as continuous "prefix" embeddings prepended to the actual input sequence embeddings $E(x_{<T})$ before they are processed by the transformer layers. The model learns to interpret this prefix as the conditioning signal specifying the target distribution. Mathematically, the input embedding sequence to the transformer becomes $[p; E(x_{<T})]$. The attention mask is adjusted accordingly to allow all sequence tokens $x_{<T}$ to attend to the prefix p .
3. **Fine-tuning:** The pre-trained generator weights can be either frozen or fine-tuned. Fine-tuning the entire model allows the LM to adapt its generation process based on the conditioning prefix p . Freezing the LM backbone and only training the conditioning projection W_p (and potentially adapter layers) can be more parameter-efficient.

A.3 Training Details and Considerations

A.3.1 Learning Rate Schedule

For simpler models we use a fixed learning rate, but for more complex models we typically employ a cosine annealing learning rate schedule during training. This involves starting with an initial learning rate and gradually decreasing it towards zero following a cosine curve over the course of training epochs. This schedule is often effective in achieving stable convergence and good final performance. In general we have found that whatever the current state of the art for training the (unconditional) generator is, that will generally give good results when learning the encoder-generator jointly.

A.3.2 Performance and Convergence

Our experiments generally indicate that this training setup, combined with the described architectures and adaptation strategies, leads to strong performance across various tasks and datasets presented in the main paper. As noted in subsection [A.1.5](#), the choice of encoder can impact convergence speed and computational cost.

A.3.3 Set Size and Batching Trade-offs

We observe that achieving optimal performance sometimes necessitates using large input set sizes (N). However, processing large sets can significantly increase the computational and memory requirements per batch, particularly for the attention mechanisms in transformer-based encoders or generators. This often forces a reduction in the overall batch size to fit within hardware constraints. Smaller batch sizes can, in turn, lead to increased variance in the loss gradients, potentially slowing down or destabilizing training. Careful tuning of the set size N , batch size, and learning rate parameters is often required to balance performance and training efficiency for a given task and hardware setup.

A.3.4 Gradient Propagation Challenges

A potential challenge arises, particularly with deeper encoder and generator architectures. The encoder only receives a learning signal indirectly through the generator via the shared latent variable z . If the generator itself struggles to utilize the latent information effectively, or if the dimensionality L of z creates an information bottleneck, the gradients flowing back to the encoder can become weak or noisy. This can make training deep encoders difficult. Addressing this might require more sophisticated generator architectures capable of integrating the latent information more effectively or alternative training schemes with auxiliary losses directly on the encoder. We found these issues in the simple encoder architectures, but they seemed to be alleviated in the ResNet-based architectures.

762 B Experiments

763 B.1 Additional semi-synthetic experimental results

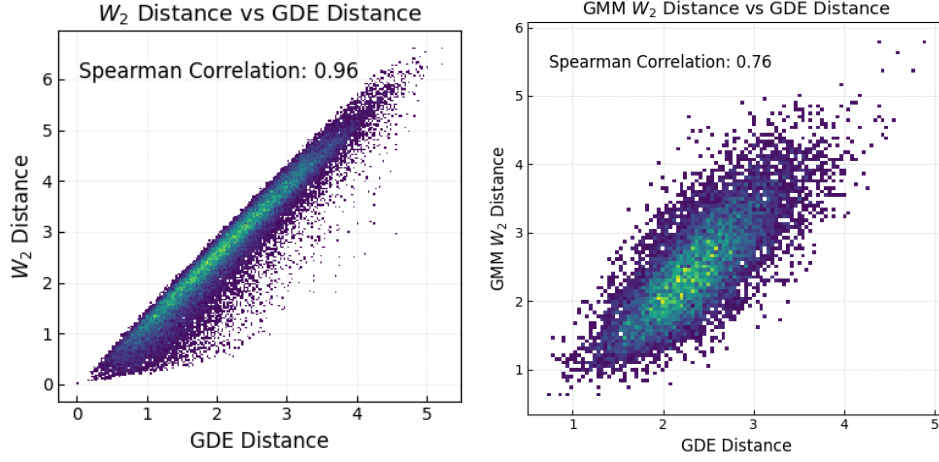


Figure 10: Left: Distance correlation showing high alignment between latent GDE distances and analytical W_2 distances (Spearman $\rho = 0.96$). Left: Distance correlation showing high alignment between latent GDE distances and the OT-GMM distance [35], which is a W_2 metric restricted to the subspace of GMMs (Spearman $\rho = 0.76$).

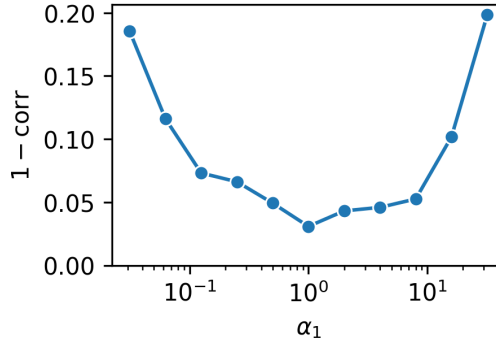


Figure 11: Expanding on Fig. 5 we show that the Pearson correlation between the W_2 (computed via normal approximation) and the latent GDE distances decreases as α_1 deviates from 1, while keeping fixed $\alpha_2 = \alpha_3 = 1$.

Table 3: W_2 reconstruction error of 30 possible GDE implementations (including two existing methods generalized by GDE, Wasserstein Wormhole and kernel mean embeddings) on 5-dimensional multivariate Gaussians. Covariance matrices sampled from Wishart distribution with scale of 1, and means sampled uniformly from $[0, 5]$. Further results included in Table 4.

Gen. \downarrow \ Enc. \rightarrow	Mean	Kernel mean	GNN	Med.-GNN	ResNet-GNN	SelfAttn.
Sinkhorn	0.05	0.14	0.09	0.10	0.05	0.06
Sliced W_2	0.03	0.04	0.07	0.07	0.03	0.04
CVAE	0.16	0.16	0.19	0.20	0.15	0.17
DDPM	0.03	0.04	0.06	0.05	0.02	0.07
Wormhole	0.14	0.15	0.72	0.49	0.14	0.20

Table 4: W_2 reconstruction error (mean \pm s.e.m. over 5 trials) for 30 possible GDE implementations (including two existing methods generalized by GDE, Wasserstein Wormhole and kernel mean embeddings) on 5-dimensional multivariate Gaussians. Covariance matrices sampled from Wishart distribution with scale of 0.1, and means sampled uniformly from $[0, 5]$.

Gen. \downarrow \ Enc. \rightarrow	Kernel mean	GNN	ResNet-GNN	Self-Attn.
CVAE	0.15 ± 0.011	0.12 ± 0.006	0.12 ± 0.009	0.11 ± 0.007
DDPM	0.15 ± 0.008	0.13 ± 0.020	0.09 ± 0.003	0.10 ± 0.005
Direct SW	0.15 ± 0.008	0.13 ± 0.007	0.13 ± 0.009	0.15 ± 0.001
Direct Sinkhorn	0.29 ± 0.008	0.22 ± 0.010	0.17 ± 0.005	0.19 ± 0.010
Wormhole	0.23 ± 0.021	0.72 ± 0.090	0.24 ± 0.011	0.34 ± 0.021

764 B.2 Lineage-traced scRNA-seq experiments

765 B.2.1 Data preprocessing details

766 We use lineage tracing data from Weinreb et al. [40]. The single-cell RNA sequencing (scRNA-seq)
 767 count matrices were preprocessed following standard procedures. Specifically, counts for each cell
 768 were normalized by rescaling to 10^4 counts per cell, followed by log transformation. Finally, the
 769 top 10^4 highly variable genes (HVGs) were selected. Cell-type annotations and two-dimensional
 770 SPRING embeddings were obtained directly from the annotations provided in Weinreb et al.

771 B.2.2 Mutual information estimation

772 We compute mutual information as a sample mean of pointwise mutual information estimates. To
 773 estimate pointwise mutual information in the representation space, we use the nonparametric nearest-
 774 neighbor estimator introduced by Kraskov et al. [63] with $k = 3$. This estimator has been shown to
 775 be effective in this setting: model latent spaces with tens of dimensions [41].

776 B.2.3 GDE modelling architecture

777 We use a Resnet-GNN architecture as the encoder and a CVAE as the generator. We use 64 latent
 778 dimensions, with 2 hidden layers of size 128.

779 B.3 Perturbation Prediction

780 B.3.1 Data preprocessing details

781 We use the pre-processed h5ad file from [46] including 10^4 genes. We compute the 10% most
 782 perturbative perturbations by examining the differentially expressed genes and then randomly select
 783 20 of those perturbations to hold out. We hold these out across all cell types.

784 B.3.2 GDE modelling architecture

785 We use a Resnet-GNN architecture as the encoder and a CVAE as the generator, similar to the
 786 architecture in the lineage-tracing experiment, except we use a larger hidden state (1024) and a
 787 larger latent space (256). We include a perturbation prediction loss during training which trains a
 788 linear model with pairwise interactions between the control cell distribution embedding and the gene
 789 embedding to predict the difference in mean expression through a linear head. This structures the
 790 latent space for our downstream perturbation prediction task.

791 B.3.3 Perturbation Prediction

792 We fit a ridge regression to predict (1) the difference in mean expression and (2) the difference between
 793 the perturbed embedding and the control for each perturbation using GenePT gene embeddings [47]
 794 with cross-validation to perform grid search over λ . We then compute the predictions on the held-out
 795 perturbations and use a linear head to predict the mean expression from the latent difference. Finally
 796 we compute the R^2 score and the MSE.

797 **B.4 Optical pooled screening dataset**

798 **B.4.1 Data preprocessing details**

799 We use phenotyping images with assigned perturbation barcodes from Funk et al. [48]. We analyze
800 only two of the measured channels: DAPI and GFP. Each image is a 64x64 bounding box surrounding
801 a single cell (center-padded or center-cropped from the original bounding box as necessary). Image
802 intensities are normalized to a minimum of -1 and a maximum of 1 . Using the set of perturbative
803 perturbations computed in [48] we randomly select 30% to holdout during training for evaluation.

804 **B.4.2 GDE modelling architecture**

805 For the encoder architecture, we extend our GNN approach to 2D convolutional layers, standard for
806 image processing. For the generator we use a U-net architecture standard in diffusion for images, but
807 upscaled in expressivity relative to our MNIST and Fashion-MNIST examples.

808 **B.4.3 Perturbation Prediction**

809 We find that empirically, our diffusion approach struggles to model the padded border of the cells.
810 So, at inference time we condition on the border to generate our predictions. Using GenePT, we
811 train a ridge regression with grid search (similar to App. B.3) to predict the perturbation distribution
812 embeddings. We also construct a nearest neighbor model using the GenePT embeddings to sample
813 the padding. We then condition on the padding and the predicted latent to sample a set of 1,000 cells
814 from each heldout perturbation. We then compute the DAPI intensity and compare with the ground
815 truth, computing the R^2 and the MSE.

816 **B.5 Methylation atlas of human tissues**

817 **B.5.1 Simulating raw bisulfite-sequencing reads from methylation patterns**

818 While sample-specific methylation patterns are published in [49], the raw sequencing reads are not
819 public due to patient privacy considerations. Here, we instead use the published methylation patterns
820 (in the form of .pat files) to simulate bisulfite sequencing reads. For each methylation site entry of
821 the .pat file, we use wgbstools[64] to find the 100 preceding bases of the HG38 genome reference,
822 and append to the CpG sequence. We omit all CpG sites with unknown methylation status. We
823 subsample 10^7 sequencing reads per sample.

824 **B.5.2 GDE modelling architecture**

825 We use a 1D convolutional neural network as our encoder, with mean pooling at each layer (analogous
826 to the fully connected GNN with an MLP, but using convolutional layers). For the generator, we
827 use HyenaDNA [30]. We additionally include a linear classification head on top of the distribution
828 embedding, co-trained with a cross-entropy loss.

829 **B.6 GPRA**

830 **B.6.1 Data processing details**

831 We collect all sequences in the Gal and Gly conditions from [50] and process them into 100 quantiles
832 by measured expression, totaling 34 million sequences. We one-hot encode these sequences for
833 ACTGN, and tokenize them using the HyenaDNA tokenizer. We break these sequences into 100
834 quantiles and hold out the top 5 quantiles during training. During training, we construct sets by
835 selecting a “center” quantile and then randomly sampling from that quantile and the two adjacent
836 quantiles.

837 **B.6.2 GDE modelling architecture**

838 We use the same architecture as in the methylation experiment (App. B.5).

839 **B.6.3 Details for Fig. 8**

840 We encode a random subsample of 130K sequences from each quantile in the Gal condition to
841 construct the set embeddings (the larger dots). We then compute the PCA of these embeddings. We
842 embed all the DNA sequences as sets of size one and project them to the PCA. For the histograms of
843 the TFBS motifs we leverage the PWMs from [51]. We wrote a simple unidirectional motif scanning
844 procedure in Torch to facilitate efficient scanning, and used a threshold of 5 to determine hits. We
845 then sum over the motifs to derive the motif count per sequence, and then compute the histogram by
846 plotting the distribution of these counts by quantile.

847 **B.7 Spatiotemporal distribution of viral lineages**

848 **B.7.1 Data preprocessing details**

849 We obtain all SARS-CoV2 spike sequences deposited up to April 2025 in GISAID [56]. We group
850 sequences by submission month and lab of collection. We discard sequences with improperly
851 formatted date fields. During tokenization, we truncate sequences to 1000 amino acids.

852 **B.7.2 GDE modelling architecture**

853 The encoder couples the ESM-50M [54] architecture coupled to a mean-pooled GNN, while the
854 generator uses the Progen2-150M architecture [29] with prefix conditioning. We initialize (but do not
855 freeze) the protein language models with their pretrained weights. We use a 128 dimensional latent
856 space.

C Background

C.1 Frequentist, Bayesian, and Predictive Sufficiency

Sufficiency is a classical notion in statistics that formalizes when a statistic retains all information about a parameter or distribution. In this appendix, we distinguish three forms of sufficiency relevant to modern generative modeling and provide canonical examples.

C.1.1 Frequentist Sufficiency

Let $\{P_\theta : \theta \in \Theta\}$ be a parametric family of probability distributions on a sample space \mathcal{X} . A statistic $T(X_1, \dots, X_n)$ is *frequentist sufficient* for θ if the conditional distribution of the data given T does not depend on θ :

$$P_\theta(X_1, \dots, X_n \mid T(X_1, \dots, X_n)) = (\text{independent of } \theta).$$

Intuitively, the likelihood depends on the data only through T .

C.1.2 Bayesian Sufficiency

Given a prior $\pi(\theta)$ over the parameter space, a statistic T is *Bayesian sufficient* for θ if the posterior depends on the data only through T :

$$\pi(\theta \mid X_1, \dots, X_n) = \pi(\theta \mid T(X_1, \dots, X_n)).$$

Bayesian sufficiency holds if and only if T is a sufficient statistic in the sense that the posterior is conditionally independent of the data given T .

C.1.3 Predictive Sufficiency

A weaker notion, often relevant in nonparametric and distributional settings, is *predictive sufficiency*. A statistic T is predictive sufficient if the distribution of a new sample X_{new} given T is the same as given the full data:

$$\mathbb{P}(X_{\text{new}} \in B \mid T(X_1, \dots, X_n)) = \mathbb{P}(X_{\text{new}} \in B \mid X_1, \dots, X_n), \quad \forall B \in \mathcal{B}(\mathcal{X}).$$

This requires only that T contains enough information to match the predictive distribution of future data.

C.1.4 Implications and Comparisons

There is a strict hierarchy among these definitions:

$$\text{Frequentist sufficiency} \Rightarrow \text{Bayesian sufficiency} \Rightarrow \text{Predictive sufficiency}.$$

The first implication follows from the factorization of the likelihood, and the second follows because the posterior predictive is a marginal of the posterior. However, the reverse implications do not hold in general, especially in infinite-dimensional or nonparametric models. In particular, predictive sufficiency may hold in settings where no finite-dimensional parameter exists.

C.1.5 Examples

Example 1 (Gaussian Mean). Let $X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma^2)$ with known σ^2 . Then the sample mean \bar{X}_n is sufficient for μ in all three senses: frequentist, Bayesian, and predictive. The likelihood, posterior, and predictive distributions all depend on the data only through \bar{X}_n .

Example 2 (Gaussian Mixture Model). Let $X_1, \dots, X_n \sim P$ where P is a finite mixture of Gaussians:

$$P = \sum_{k=1}^K \pi_k \mathcal{N}(\mu_k, \Sigma_k).$$

The sufficient statistics for this model (under known K) are:

- the soft assignment (responsibility) weights for each component,

- the empirical means and covariances of points assigned to each component,
- the mixture proportions.

These are sufficient in both the frequentist and Bayesian senses. In many applications, they are approximated via the Expectation-Maximization algorithm or variational inference.

Example 3 (Uniform(0, θ)). Let $X_1, \dots, X_n \sim \text{Unif}(0, \theta)$. Then the sample maximum

$$T_n = \max\{X_1, \dots, X_n\}$$

is the minimal sufficient statistic for θ in both the frequentist and Bayesian senses. It also suffices for prediction of future samples, since the predictive distribution under θ is uniform on $[0, \theta]$, and T_n provides all information about θ .

C.1.6 Nonparametric Extensions

In the nonparametric regime where P is not indexed by a finite-dimensional parameter, predictive sufficiency remains well-defined. For instance, the empirical measure $P_n = \frac{1}{n} \sum_{i=1}^n \delta_{X_i}$ is always predictive sufficient under exchangeable models. In this setting, stronger forms of sufficiency may not exist, but predictive sufficiency still supports meaningful generative modeling.

C.2 Otto's Geometry and Statistical Submanifolds

This appendix summarizes the formal Riemannian structure of the Wasserstein space $\mathcal{P}_2(\mathcal{X})$ introduced by Otto [12], and defines statistical manifolds as submanifolds equipped with a geometry induced from this structure. This provides the mathematical foundation for interpreting generative distributional encoders (GDEs) as learning smooth geometric embeddings of constrained distributional families.

C.2.1 Wasserstein Space and the Benamou–Brenier Formulation

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a domain, and let $\mathcal{P}_2(\mathcal{X})$ denote the space of Borel probability measures on \mathcal{X} with finite second moment. The 2-Wasserstein distance between two measures $\mu_0, \mu_1 \in \mathcal{P}_2(\mathcal{X})$ is defined by the optimal transport problem

$$W_2^2(\mu_0, \mu_1) := \inf_{\gamma \in \Pi(\mu_0, \mu_1)} \int_{\mathcal{X} \times \mathcal{X}} \|x - y\|^2 d\gamma(x, y),$$

where $\Pi(\mu_0, \mu_1)$ denotes the set of couplings with marginals μ_0 and μ_1 . An equivalent dynamic formulation, due to Benamou and Brenier [65], expresses the Wasserstein distance as a variational problem over time-dependent flows:

$$W_2^2(\mu_0, \mu_1) = \inf_{\substack{(\mu_t, v_t) \\ \partial_t \mu_t + \nabla \cdot (\mu_t v_t) = 0}} \int_0^1 \int_{\mathcal{X}} \|v_t(x)\|^2 d\mu_t(x) dt,$$

subject to boundary conditions μ_0, μ_1 and the continuity equation, which ensures mass conservation.

C.2.2 Otto's Riemannian Structure

Otto observed that the Benamou–Brenier problem defines a formal Riemannian structure on $\mathcal{P}_2(\mathcal{X})$, where the tangent space at a measure μ consists of velocity fields v such that the continuity equation describes admissible perturbations. The inner product between two such velocity fields $v_1, v_2 \in T_\mu \mathcal{P}_2$ is defined as

$$\langle v_1, v_2 \rangle_{T_\mu \mathcal{P}_2} := \int_{\mathcal{X}} v_1(x) \cdot v_2(x) d\mu(x).$$

This makes $\mathcal{P}_2(\mathcal{X})$ a formal infinite-dimensional Riemannian manifold, and Wasserstein geodesics become curves of minimal kinetic energy under this metric.

For absolutely continuous μ_0 , the optimal transport map $T : \mathcal{X} \rightarrow \mathcal{X}$ from μ_0 to μ_1 induces a geodesic $(\mu_t)_{t \in [0, 1]}$ by pushing μ_0 along linear interpolations:

$$\mu_t = ((1 - t)\text{id} + tT)_{\#} \mu_0.$$

These displacement interpolants travel at constant speed under the W_2 metric and solve the geodesic equation associated with the Otto metric.

930 **C.2.3 Statistical Manifolds as Submanifolds of Wasserstein Space**

931 Let Q be a probability distribution over $\mathcal{P}_2(\mathcal{X})$, and define the *statistical manifold* as the support of
 932 Q :

$$\mathcal{M} := \text{supp}(Q) \subset \mathcal{P}_2(\mathcal{X}).$$

933 We endow \mathcal{M} with the *induced Riemannian structure* from $\mathcal{P}_2(\mathcal{X})$, by restricting the Otto metric
 934 to velocity fields that remain tangent to \mathcal{M} . That is, $T_\mu \mathcal{M} \subset T_\mu \mathcal{P}_2$ is a subspace of velocity fields
 935 preserving membership in \mathcal{M} , and the inner product is

$$\langle v_1, v_2 \rangle_{T_\mu \mathcal{M}} := \int_{\mathcal{X}} v_1(x) \cdot v_2(x) d\mu(x), \quad \text{for } v_1, v_2 \in T_\mu \mathcal{M}.$$

936 This leads to a constrained transport problem defining geodesics within \mathcal{M} :

$$W_{2,\mathcal{M}}^2(\mu_0, \mu_1) := \inf_{\substack{(\mu_t, v_t) \\ \mu_t \in \mathcal{M}, \partial_t \mu_t + \nabla \cdot (\mu_t v_t) = 0}} \int_0^1 \int_{\mathcal{X}} \|v_t(x)\|^2 d\mu_t(x) dt.$$

937 This is simply the Wasserstein variational problem, but restricted to paths that lie within the submani-
 938 fold \mathcal{M} . It defines the geometry relevant to learning distributions drawn from Q .

939 **C.2.4 Examples and Application to GDEs**

940 Typical examples of such submanifolds include:

- 941 • Gaussian families $\mathcal{N}(\mu, \Sigma)$, where geodesics can be computed in closed form;
- 942 • Mixture models with a fixed number of components, see [\[35\]](#);
- 943 • general parametric families.

944 In this work, we treat the statistical manifold $\mathcal{M} = \text{supp}(Q)$ as the set of data-generating distributions
 945 and interpret GDEs as learning a smooth embedding of this submanifold into Euclidean space. While
 946 GDEs do not explicitly minimize Wasserstein distances, we observe empirically that their learned
 947 latent geometries often approximate the structure of $W_{2,\mathcal{M}}$, suggesting that they act as approximate
 948 isometric embeddings of this constrained transport geometry.

949 D Theory

950 Throughout, let $(\mathcal{X}, \mathcal{B})$ be a Polish space. Let $P \in \mathcal{P}(\mathcal{X})$ denote a probability law on \mathcal{X} . Given
 951 $m \in \mathbb{N}$, let $S_m = (X_1, \dots, X_m)$ be an i.i.d. sample from P , and let $P_m = \frac{1}{m} \sum_{i=1}^m \delta_{X_i}$ denote the
 952 empirical measure.

953 We use P_1, P_2 to denote two (possibly distinct) probability laws on \mathcal{X} , and S_1, S_2 for independent
 954 samples from P_1, P_2 respectively.

955 For signed measures ν, μ on $(\mathcal{X}, \mathcal{B})$ define

$$d_{\text{BL}}(\nu, \mu) := \sup_{\substack{f: \mathcal{X} \rightarrow [-1, 1] \\ \text{Lip}(f) \leq 1}} \left| \int f d(\nu - \mu) \right|.$$

956 We use $\|\cdot\|_{\text{BL}}$ for the corresponding norm $\|\nu\|_{\text{BL}} := d_{\text{BL}}(\nu, 0)$ and recall that $d_{\text{TV}}(\nu, \mu) \leq$
 957 $d_{\text{BL}}(\nu, \mu)$.

958 All random variables are defined on a common probability space unless otherwise specified.

959 D.1 Necessity of Distributional Invariance

960 **Motivation** Our goal is to design encoder architectures that flexibly model unknown data distribu-
 961 tions while guaranteeing consistent generation of the underlying law as sample size grows. Since the
 962 true distribution P is not known in advance, the encoder must be constructed to generalize across all
 963 possible P , without leaking spurious information tied to the specific realization or sample size. If the
 964 encoder depends on sample-level artifacts—such as ordering, multiplicity, or the raw sample size—it
 965 may encode features that a generator can exploit, breaking the guarantee that

$$\mathcal{G}(\mathcal{E}(S_m)) \xrightarrow{d} P \quad \text{as } m \rightarrow \infty, \quad S_m \sim P^{\otimes m}.$$

966 This risk arises even under either permutation or proportional invariance on their own: both permit
 967 dependencies that vanish only in expectation and are insufficient to ensure correct extrapolation with
 968 increasing m . For example, encoders based on unnormalized sum aggregations (e.g., DeepSets) will
 969 vary with m even when the empirical distribution is unchanged, leading to divergence at inference
 970 time.

971 To formalize this constraint, we draw on Blackwell’s theory of experiments, which provides a general
 972 framework for comparing the informativeness of statistical summaries. We adopt his game-theoretic
 973 perspective—viewing the encoder as a player that chooses an experiment, and the generator as an
 974 adversary that exploits the information it receives—and use this to characterize the minimal structural
 975 conditions an encoder must satisfy to guarantee asymptotic consistency. In particular, we show
 976 that dependence on the empirical distribution is necessary and sufficient: it is the least informative
 977 summary that still retains all information required to identify the law, so the generator cannot learn
 978 any spurious information that will fail to extrapolate at inference time.

979 **Setting** We consider the following general setting: Let $(\mathcal{X}, \mathcal{B})$ be a Polish space. We are interested
 980 in measurable summaries of infinite i.i.d. sequences $S \sim P^\infty$, where $P \in \mathcal{P}(\mathcal{X})$ is an unknown
 981 probability law. The goal is to characterize the minimal invariance properties required for encoders to
 982 guarantee consistent recovery of P from finite samples.

983 **Definition 1** (Distributional Invariance). A function $\mathcal{E} : \mathcal{X}^m \rightarrow \mathcal{Z}$ is *distributionally invariant* if for
 984 any $S_m \in \mathcal{X}^m$, $\mathcal{E}(S_m)$ depends only on the empirical measure P_m of S_m ; that is, for any permutation
 985 π of $\{1, \dots, m\}$ and any S_m , $\mathcal{E}(S_m) = \mathcal{E}(S_m^\pi)$, and ϕ is invariant to proportional duplications of the
 986 sample.

987 **Definition 2** (Asymptotic Distributional Invariance). A sequence of functions $\mathcal{E}_m : \mathcal{X}^m \rightarrow \mathcal{Z}$ is
 988 *asymptotically distributionally invariant* if for every $P \in \mathcal{P}(\mathcal{X})$, there exists a sequence of measurable
 989 functions $\phi_m : \mathcal{P}_m(\mathcal{X}) \rightarrow \mathcal{Z}$ such that

$$\mathbb{P}_{S_m \sim P^{\otimes m}}(\mathcal{E}_m(S_m) = \phi_m(P_m)) \rightarrow 1 \quad \text{as } m \rightarrow \infty,$$

990 where P_m is the empirical measure of S_m .

991 **Lemma 1** (Strong Law of Large Numbers for Empirical Measures). *Let $P \in \mathcal{P}(\mathcal{X})$ and $S_m =$*
 992 *(X_1, \dots, X_m) be i.i.d. samples from P . Then the empirical measure $P_m = \frac{1}{m} \sum_{i=1}^m \delta_{X_i}$ converges*
 993 *almost surely to P in the weak topology as $m \rightarrow \infty$.*

994 **Lemma 2** (Wainwright’s Rademacher–tail bound [66, Thm. 4.10]). *Let \mathcal{F} be a class of measurable*
 995 *functions $f : \mathcal{X} \rightarrow \mathbb{R}$ that is b -uniformly bounded, i.e. $\|f\|_\infty \leq b$ for all $f \in \mathcal{F}$. For any integer*
 996 *$n \geq 1$ and any $\delta > 0$,*

$$\|P_n - P\|_{\mathcal{F}} \leq 2 \mathcal{R}_n(\mathcal{F}) + \delta \quad \text{with probability at least } 1 - \exp\left(-\frac{n\delta^2}{2b^2}\right),$$

997 *where $\mathcal{R}_n(\mathcal{F})$ is the (deterministic) Rademacher complexity defined in [66, Eq. 4.13].*

998 To formalize what can go wrong, we introduce an adversarial two-player game, adapted from the
 999 decision-theoretic framework introduced by [67]. This game reveals the informational limits of
 1000 summary statistics for distinguishing distributions.

- 1001 1. *Player 1* selects a measurable summary rule $T : \mathcal{X}^{\mathbb{N}} \rightarrow \mathcal{T}$ before seeing any data.
- 1002 2. *Player 2* observes T and chooses two probability laws $P_1, P_2 \in \mathcal{P}(\mathcal{X})$.
- 1003 3. *Nature* draws two independent infinite i.i.d. sequences $S_1 \sim P_1^\infty, S_2 \sim P_2^\infty$.

1004 The induced decision problem is whether the summary T is consistent with $P_1 = P_2$ or not. The
 1005 payoff structure is:

	$T(S_1) = T(S_2)$	$T(S_1) \neq T(S_2)$
$P_1 = P_2$	(1, 0)	(0, 1)
$P_1 \neq P_2$	(0, 1)	(1, 0)

1006 Player 1 aims to minimize both types of errors: introducing spurious distinctions when $P_1 = P_2$, and
 1007 failing to distinguish when $P_1 \neq P_2$. In Blackwell’s terms, the goal is to find a summary that is as
 1008 informative as possible for this class of binary decision problems.

1009 **Definition 3** (Asymptotically Blackwell–optimal summary). Let $(\mathcal{T}, d_{\mathcal{T}})$ be a separable metric space
 1010 and let $T_n : \mathcal{X}^n \rightarrow \mathcal{T}$ be measurable. Fix any deterministic sequence $\varepsilon_n \downarrow 0$. We say that (T_n) is
 1011 *asymptotically Blackwell–optimal* if, for every pair of probability laws $P_1, P_2 \in \mathcal{P}(\mathcal{X})$,

$$(i) \quad P_1 = P_2 \implies \mathbb{P}_{P_1^\infty} \left[d_{\mathcal{T}}(T_n(S_1), T_n(S_2)) > \varepsilon_n \right] \xrightarrow{n \rightarrow \infty} 0, \quad (2)$$

$$(ii) \quad P_1 \neq P_2 \implies \mathbb{P}_{P_1^\infty \times P_2^\infty} \left[d_{\mathcal{T}}(T_n(S_1), T_n(S_2)) \leq \varepsilon_n \right] \xrightarrow{n \rightarrow \infty} 0. \quad (3)$$

1012 Main Result

1013 **Theorem 1** (Empirical distribution characterises optimal asymptotic summaries). *Let $P_n(S) =$*
 1014 *$\frac{1}{n} \sum_{i=1}^n \delta_{S_i}$ be the empirical distribution of the first n samples of $S \sim P^\infty$. For measurable*
 1015 *summaries $T_n : \mathcal{X}^n \rightarrow \mathcal{T}$:*

- 1016 (i) **(Blackwell optimality)** *the empirical distribution $T_n^{\text{emp}}(S) = P_n(S)$ is asymptotically optimal*
 1017 *in the game of Definition 3*
- 1018 (ii) **(Asymptotic information equivalence)** *If (T_n) is asymptotically optimal, there exist measurable*
 1019 *maps $f_n : \mathcal{P}(\mathcal{X}) \rightarrow \mathcal{T}$ and $g_n : \mathcal{T} \rightarrow \mathcal{P}(\mathcal{X})$ such that for every $P \in \mathcal{P}(\mathcal{X})$*

$$\mathbb{P}_{P^{\otimes n}}(T_n(S) = f_n(P_n(S))) \xrightarrow{n \rightarrow \infty} 1, \quad \mathbb{P}_{P^{\otimes n}}(P_n(S) = g_n(T_n(S))) \xrightarrow{n \rightarrow \infty} 1.$$

1020 *In particular, T_n and P_n are asymptotically Blackwell-equivalent in probability.*

1021 We can immediately conclude the implications for encoders here:

1022 **Corollary 1** (Necessity of asymptotic distributional sufficiency). Suppose we design an encoder \mathcal{E}
 1023 and decoder \mathcal{G} with the goal that, for any unknown distribution $P \in \mathcal{P}(\mathcal{X})$,

$$\mathcal{G}(\mathcal{E}(S_m)) \xrightarrow{d} P \quad \text{as } m \rightarrow \infty, \quad S_m \sim P^{\otimes m}.$$

1024 Then, in order for this convergence to hold for any P , the encoder architecture must satisfy two
 1025 conditions:

1026 (i) **Asymptotic distributional invariance:** $\mathcal{E}_m(S_m)$ must (eventually) depend only on the
 1027 empirical distribution $P_m = \frac{1}{m} \sum_{j=1}^m \delta_{x_j}$ —that is, for every P , $\mathbb{P}_{P^{\otimes m}}(\mathcal{E}_m(S_m) =$
 1028 $\phi_m(P_m)) \rightarrow 1$.

1029 (ii) **Distributional expressivity:** the class of encoder functions must be rich enough to approxi-
 1030 mate any measurable function of P_m .

1031 These are constraints on the encoder *architecture*, not on the learned function after training. If \mathcal{E}_m
 1032 encodes any features not measurable with respect to P_m —such as sample order, indexing artifacts,
 1033 or features sensitive to repeated observations—the generator can exploit these to fit P incorrectly,
 1034 breaking consistency. Either permutation or proportional invariance alone are not sufficient: only
 1035 sufficiency with respect to the empirical distribution rules out such failure modes.

1036 **Proof**

1037 *Proof. Step 1: Soundness.* The class \mathcal{F}_{BL} is 1-uniformly bounded, and its empirical Rademacher
 1038 complexity satisfies $\mathfrak{R}_n(\mathcal{F}_{\text{BL}}) = O(n^{-1/2})$. Equip \mathcal{T} with d_{BL} and set $\varepsilon_n = n^{-1/2}$.

1039 By the triangle inequality,

$$d_{\text{BL}}(P_n(S_1), P_n(S_2)) \leq d_{\text{BL}}(P_n(S_1), P) + d_{\text{BL}}(P_n(S_2), P).$$

1040 Applying Lemma 2 [66] to each term with tolerance $\varepsilon_n/2$ and union-bounding yields

$$\mathbb{P}\left[d_{\text{BL}}(P_n(S_1), P_n(S_2)) > \varepsilon_n\right] \leq 2 \exp\left(-\frac{n\varepsilon_n^2}{8}\right).$$

1041 Choosing $\varepsilon_n = n^{-1/4}$ therefore fulfils (2).

1042 If $P_1 \neq P_2$, the strong law gives $d_{\text{BL}}(P_n(S_i), P_i) \xrightarrow{\text{a.s.}} 0$, hence for large n the event
 1043 $d_{\text{BL}}(P_n(S_1), P_n(S_2)) \leq \varepsilon_n$ is impossible, establishing (3).

1044 **Step 2: T_n is a function of P_n with high probability.** Let $\eta_n \downarrow 0$ be an arbitrary deterministic
 1045 sequence. We will show

$$\mathbb{P}_{P^{\otimes n}}[T_n(S) = f_n(P_n(S))] \geq 1 - \eta_n \quad \text{for all large } n,$$

1046 for some measurable $f_n: \mathcal{P}(\mathcal{X}) \rightarrow \mathcal{T}$. That is, T_n differs from a measurable function of the empirical
 1047 distribution with probability $o(1)$, which suffices for the asymptotic game.

1048 Now fix $\eta > 0$ and define

$$\mathcal{I}_\eta := \left\{ n \geq 1 : \exists P \in \mathcal{P}(\mathcal{X}) \text{ s.t. } \mathbb{P}_{P^{\otimes n}}[T_n(S) \text{ is not } \sigma(P_n)\text{-measurable}] > \eta \right\}.$$

1049 If \mathcal{I}_η were infinite for some positive η , we would construct a single distribution P_\dagger to be any
 1050 accumulation point of the sequence of counter-example measures forcing

$$\mathbb{P}[T_n(S_1) \neq T_n(S_2)] \geq \frac{\eta^2}{2} \quad \text{for all } n \in \mathcal{I}_\eta,$$

1051 contradicting optimality condition (2).

1052 Since asymptotic optimality holds, every $\eta > 0$ gives a finite \mathcal{I}_η , so we can choose $N(\eta)$ such that

$$\mathbb{P}_{P^{\otimes n}}[T_n(S) \text{ is } \sigma(P_n)\text{-measurable}] \geq 1 - \eta \quad \text{for all } n \geq N(\eta) \text{ and every } P.$$

1053 Taking $\eta = \eta_n$ and letting f_n be any measurable selector on the high-probability event (where T_n is
 1054 $\sigma(P_n)$ -measurable) proves the claim that T_n is *eventually* a function of P_n with probability $1 - \eta_n$.

1055 **Step 3: Recoverability of P_n from T_n .** If f_n collapses two distinct empirical measures $m \neq m'$,
 1056 take $P_1 = m$, $P_2 = m'$. Then $d_{\text{BL}}(m, m') > 0$ while $T_n(S_1) = T_n(S_2)$ a.s., violating (3). Hence
 1057 there exists a measurable $g_n: \mathcal{T} \rightarrow \mathcal{P}(\mathcal{X})$ with $P_n = g_n(T_n)$ a.s. \square

1058 **Remark.** The argument above shows that any asymptotically optimal summary T_n is both a measur-
 1059 able function of P_n and sufficient to recover P_n almost surely. In Blackwell’s terminology, this means
 1060 T_n and P_n are asymptotically equivalent experiments: they contain exactly the same information for
 1061 distinguishing distributions from i.i.d. samples. In future work it would be interesting to consider
 1062 how to design architectures with similar properties under different forms of dependence.

1063 D.2 A Complete Large- m Analysis of the Plug-in Loss

1064 **Motivation** We analyze the statistical properties of the plug-in loss used to train distributional
 1065 encoders and generators. Our goal is to understand the asymptotic behavior of this loss as the sample
 1066 size grows, and to establish conditions under which the learned generator recovers the true data
 1067 distribution. This analysis provides a principled foundation for the training objectives used in our
 1068 framework.

1069 **Setting** First we establish some notation and definitions.

1070 **Definition 4** (Hadamard differentiability). A map $T : \mathcal{D} \rightarrow \mathcal{Y}$ between normed spaces is *Hadamard*
 1071 *differentiable* at $x \in \mathcal{D}$ if there exists a continuous linear operator DT_x such that for every sequence
 1072 $h_t \rightarrow h$ in \mathcal{D} and $t \downarrow 0$, $\frac{T(x+th_t) - T(x)}{t} \rightarrow DT_x[h]$.

1073 **Definition 5** (Fréchet differentiability). Let $T : \mathcal{D} \rightarrow \mathcal{Y}$ be a map between normed vector spaces. T
 1074 is *Fréchet differentiable* at $x \in \mathcal{D}$ if there exists a bounded linear operator $A : \mathcal{D} \rightarrow \mathcal{E}$ such that

$$\lim_{\|h\|_{\mathcal{D}} \rightarrow 0} \frac{\|T(x+h) - T(x) - A(h)\|_{\mathcal{E}}}{\|h\|_{\mathcal{D}}} = 0.$$

1075 The operator A is called the Fréchet derivative of T at x .

1076 We work in the following general setting:

1077 **Assumption 1** (Data and Empirical Measure). $(\mathcal{X}, \mathcal{B})$ is a Polish space; $P \in \mathcal{P}(\mathcal{X})$ is the true data
 1078 law. Observations $S_m = (X_1, \dots, X_m)$ are i.i.d. P . The empirical measure is $P_m = \frac{1}{m} \sum_{i=1}^m \delta_{X_i}$.

1079 **Assumption 2** (Encoder regularity). For each probability law $P \in \mathcal{P}(\mathcal{X})$ the encoder $\phi : \mathcal{P}(\mathcal{X}) \rightarrow$
 1080 \mathbb{R}^d satisfies

- 1081 (i) **Distributional invariance:** $\mathcal{E}_m(S_m) = \phi(P_m)$ depends on the sample only via its empirical
 1082 measure.
- 1083 (ii) **Pathwise (Hadamard) differentiability:** ϕ is pathwise differentiable at P and its canonical
 1084 gradient¹ $\psi_P : \mathcal{X} \rightarrow \mathbb{R}^d$ belongs to $L^2(P)$.
- 1085 (iii) **Asymptotic linearity (AL):** the estimator obeys

$$\sqrt{m} \{ \phi(P_m) - \phi(P) \} = \frac{1}{\sqrt{m}} \sum_{i=1}^m \psi_P(X_i) + o_p(1).$$

1086 where $\mathbb{E}_{X \sim P}[\psi_P(X)] = 0$, so ψ_P .

1087 Under these conditions

$$\sqrt{m} \{ \phi(P_m) - \phi(P) \} \xrightarrow{d} \mathcal{N}(0, \Sigma_\phi), \quad \Sigma_\phi := \text{Var}_{X \sim P}[\psi_P(X)].$$

1088 **Assumption 3** (Generator). $\mathcal{G} : \mathbb{R}^d \rightarrow \mathcal{P}(\mathcal{X})$ is Fréchet differentiable on a neighbourhood of
 1089 $\mu := \phi(P)$ and its derivative factors through $L^2(P)$, i.e.

$$D_\mu \mathcal{G} = T \circ A, \quad \text{where } A : \mathbb{R}^d \rightarrow L_0^2(P), \quad T : L_0^2(P) \rightarrow \mathcal{M}_0(\mathcal{X})$$

1090 are bounded linear maps and $L_0^2(P)$ denotes zero-mean square-integrable functions.

1091 **Assumption 4** (Divergence). The discrepancy $\mathfrak{d} : \mathcal{P}(\mathcal{X})^2 \rightarrow \mathbb{R}_+$ satisfies

- 1092 (i) **(Hadamard differentiability)** the map $Q \mapsto \mathfrak{d}(P, Q)$ is Hadamard differentiable at $Q_0 =$
 1093 $\mathcal{G}(\mu)$ tangentially to $\mathcal{M}_0(\mathcal{X})$, with continuous linear derivative $D_2 \mathfrak{d}(P, Q_0) : \mathcal{M}_0(\mathcal{X}) \rightarrow$
 1094 \mathbb{R} ;
- 1095 (ii) **(Separating property)** $\mathfrak{d}(P, Q) = 0 \implies P = Q$;
- 1096 (iii) **(Weak-continuity)** if $\mathfrak{d}(Q_n, Q) \rightarrow 0$ then $Q_n \Rightarrow Q$.

¹In the semiparametric sense of [68] i.e. the unique influence function representing the functional derivative along $\mathcal{M}_0(\mathcal{X})$.

1097 We define a general loss function $\ell(P, Q)$, where P is the true distribution and Q is a model output
 1098 (e.g., a divergence such as KL or Wasserstein).

1099 The *plug-in loss* is

$$\widehat{\ell}_m := \ell(P, \mathcal{G}(\phi(P_m)))$$

1100 and the *population loss* is

$$\ell^* := \ell(P, \mathcal{G}(\phi(P)))$$

1101 where P_m is the empirical measure of the sample, ϕ is the encoder, and \mathcal{G} is the generator.

1102 **Lemma 3** (Donsker's Theorem for Empirical Measures). *Let $(\mathcal{X}, \mathcal{B})$ be a Polish space, and let*
 1103 *$P \in \mathcal{P}(\mathcal{X})$. Let $\{X_i\}_{i=1}^m$ be i.i.d. samples from P , and let P_m be the empirical measure:*

$$P_m = \frac{1}{m} \sum_{i=1}^m \delta_{X_i}.$$

1104 *Define the empirical process:*

$$\sqrt{m}(P_m - P).$$

1105 *Then, viewed as an element of the Banach space $\ell^\infty(\mathcal{F})$ of bounded real-valued functions on \mathcal{F} ,*
 1106 *where \mathcal{F} is any P -Donsker class of measurable functions, we have:*

$$\sqrt{m}(P_m - P) \xrightarrow{d} \mathbb{G}_P,$$

1107 *where \mathbb{G}_P is a P -Brownian bridge, a mean-zero tight Gaussian process indexed by \mathcal{F} with covariance*
 1108 *function*

$$\text{Cov}(\mathbb{G}_P(f), \mathbb{G}_P(g)) = \text{Cov}_{X \sim P}(f(X), g(X)).$$

1109 **Lemma 4** (Functional Delta Method, [68 Thm. 3.9.4]). *Let $(\mathbb{D}, \|\cdot\|_{\mathbb{D}})$ and $(\mathbb{E}, \|\cdot\|_{\mathbb{E}})$ be normed*
 1110 *vector spaces. Let $T : \mathbb{D} \rightarrow \mathbb{E}$ be a map that is Hadamard differentiable at a point $z \in \mathbb{D}$ tangentially*
 1111 *to a subset $\mathbb{D}_0 \subseteq \mathbb{D}$, with continuous linear derivative denoted $DT_z : \mathbb{D}_0 \rightarrow \mathbb{E}$.*

1112 *Suppose:*

1113 (a) *There exist random elements Z_m taking values in \mathbb{D} such that:*

$$\sqrt{m}(Z_m - z) \xrightarrow{d} Z$$

1114 *for some tight limit Z taking values in \mathbb{D}_0 .*

1115 (b) *Z is tight and Borel measurable.*

1116 *Then:*

$$\sqrt{m}(T(Z_m) - T(z)) \xrightarrow{d} DT_z(Z),$$

1117 *where $DT_z(Z)$ is a random element of \mathbb{E} .*

1118 *In particular, if Z is Gaussian in \mathbb{D}_0 and DT_z is continuous and linear, then $DT_z(Z)$ is Gaussian in*
 1119 *\mathbb{E} .*

1120 Main Result

1121 **Theorem 2** (Large- m behaviour of the plug-in loss). *Assume [1], [2], [3], and [4]. Let $\mu := \phi(P)$ and*
 1122 *$\widehat{\ell}_m := \mathfrak{d}(P, \mathcal{G}(\phi(P_m)))$.*

1123 *We now combine the regularity assumptions with empirical-process theory to quantify the estimation*
 1124 *error of the plug-in loss*

1125 (a) **Asymptotic normality of the Encoder.**

$$\sqrt{m}\{\phi(P_m) - \phi(P)\} \xRightarrow{d} \mathcal{N}(0, \Sigma_\phi), \quad \Sigma_\phi := \text{Var}_{X \sim P}[\psi_P(X)].$$

1126 (b) **Unbiasedness of the loss.** $\mathbb{E}[\widehat{\ell}_m] = \ell^* + O(m^{-1}), \quad \ell^* := \mathfrak{d}(P, \mathcal{G}(\mu)).$

1127 (c) **Asymptotic normality of the loss.**

$$\sqrt{m}(\widehat{\ell}_m - \ell^*) \xRightarrow{d} \mathcal{N}(0, \sigma^2), \quad \sigma^2 = \nabla_\mu \ell^\top \Sigma_\phi \nabla_\mu \ell, \quad \ell(\theta) := \mathfrak{d}(P, \mathcal{G}(\theta)).$$

1128 (d) **Sufficiency of the loss.** If (ϕ^*, \mathcal{G}^*) minimises $P \mapsto \mathfrak{d}(P, \mathcal{G}(\phi(P)))$, then $\mathcal{G}^*(\phi^*(P_m)) \Rightarrow P$ in
 1129 probability as $m \rightarrow \infty$.

1130 *Proof.* **Step 1: Asymptotic Normality of the encoder (a).** Assumption 2(iii) (asymptotic linearity)
 1131 gives

$$\sqrt{m}\{\phi(P_m) - \phi(P)\} = \frac{1}{\sqrt{m}} \sum_{i=1}^m \psi_P(X_i) + o_p(1),$$

1132 and the classical multivariate CLT yields the stated convergence.

1133 Let $\Delta_m := \phi(P_m) - \mu$ so that, by (a), $\sqrt{m} \Delta_m \xrightarrow{d} \mathcal{N}(0, \Sigma_\phi)$.

1134 **Step 2 (unbiasedness).** Because $\mathbb{E}[\Delta_m] = 0$ by Assumption 2(iii) and ℓ is twice continuously
 1135 differentiable in a neighbourhood of μ , a Taylor expansion gives $\mathbb{E}[\ell_m - \ell^*] = \frac{1}{2} \text{tr}\{\nabla_\mu^2 \ell \text{Var} \Delta_m\} +$
 1136 $O(m^{-1}) = O(m^{-1})$.

1137 **Step 3: Asymptotic Normality of the loss (c).** Apply the functional delta method twice:

1138 1. to the generator $\mathcal{G} : \mathbb{R}^d \rightarrow \mathcal{P}(\mathcal{X})$, using Fréchet differentiability and the fact that $\sqrt{m} \Delta_m$ is tight,
 1139 obtaining

$$\mathcal{G}(\mu + \Delta_m) = \mathcal{G}(\mu) + D_\mu \mathcal{G}[\Delta_m] + o_p(m^{-1/2});$$

1140 2. to the divergence $Q \mapsto \mathfrak{d}(P, Q)$ at $Q_0 := \mathcal{G}(\mu)$, with linear derivative $\partial_2 \mathfrak{d}(P, Q_0)[\cdot]$.

1141 Combining the two expansions produces the linear functional of $\sqrt{m} \Delta_m$ displayed in (c) and hence
 1142 the Gaussian limit with variance $\sigma^2 = \nabla_\mu \ell^\top \Sigma_\phi \nabla_\mu \ell$.

1143 **Step 4: Sufficiency of the loss (d).** If (ϕ^*, \mathcal{G}^*) is optimal, then $\mathfrak{d}(P, \mathcal{G}^*(\phi^*(P))) = 0$,
 1144 so $\mathcal{G}^*(\phi^*(P)) = P$ by Assumption 4(ii). Repeating the expansion from (c) with (ϕ^*, \mathcal{G}^*)
 1145 shows $\mathfrak{d}(P, \mathcal{G}^*(\phi^*(P_m))) = O_p(m^{-1/2})$, and consistency for weak convergence then implies
 1146 $\mathcal{G}^*(\phi^*(P_m)) \Rightarrow P$. \square

1147 **Encoders: examples, counter-examples, and CLTs** The only encoder requirement entering
 1148 Theorem 2 is Assumption 2. We now show that it is satisfied by a large family of permutation-invariant
 1149 architectures built from *asymptotically-linear* (M/Z) *poolers*.

1150 **Generic K -layer pool-concat encoder** Fix $K \in \mathbb{N}$. Given a set of samples $S_m = \{x_1, \dots, x_m\}$
 1151 define recursively

$$h_i^{(0)} = \psi(x_i), \quad \bar{h}^{(\ell)} = T^{(\ell)}(h_{1:m}^{(\ell-1)}), \quad h_i^{(\ell)} = \text{MLP}_\ell(h_i^{(\ell-1)}, \bar{h}^{(\ell)}), \quad \ell = 1, \dots, K,$$

1152 and set the encoder output to be another pooler $\phi(P_m) = T^{(K+1)}(h_{1:m}^{(K)})$.

1153 We call a permutation-invariant functional an *asymptotically linear (AL) pooler* if it is root- m
 1154 consistent and admits an influence-function expansion; precise details follow.

1155 **Definition 6** (Asymptotically-linear pooler). A symmetric map $T : \mathcal{X}^m \rightarrow \mathbb{R}^d$ is an *AL pooler* at law
 1156 P if there exists $\psi_P \in L^2(P)$ such that

$$\sqrt{m} \{T(X_{1:m}) - \phi(P)\} = \frac{1}{\sqrt{m}} \sum_{i=1}^m \psi_P(X_i) + o_p(1).$$

1157 Examples: mean, median, trimmed mean, Huber M -estimator, M -quantiles, studentised Z -estimators
 1158 with finite variance.

1159 **Proposition 4** (CLT for K -layer AL pool-concat encoders). *Assume*

- 1160 (i) each $T^{(\ell)}$ ($\ell = 1, \dots, K + 1$) is an AL pooler at P ;
- 1161 (ii) each MLP_ℓ and the base feature map $\psi : \mathcal{X} \rightarrow \mathbb{R}^p$ are C^2 with bounded derivatives, and
 1162 weights are frozen as $m \rightarrow \infty$.

1163 Then the encoder ϕ is distributionally invariant, pathwise differentiable, and satisfies the CLT of
 1164 Assumption 2 with

$$\sqrt{m}\{\phi(P_m) - \phi(P)\} \xrightarrow{d} \mathcal{N}(0, \Sigma_\phi),$$

1165 for some finite covariance matrix Σ_ϕ .

1166 *Sketch.* The composition of Lipschitz maps (MLP_ℓ) with AL poolers is Hadamard differentiable
 1167 by repeated application of the delta method (iterating Lemma 4 [68]). Plugging each AL ex-
 1168 pansion into the chain yields an overall AL expansion whose leading empirical-process term is
 1169 $m^{-1/2} \sum_{i=1}^m \psi_P^*(X_i)$ for some $L^2(P)$ function ψ_P^* , giving the CLT. \square

1170 Instantiation to common architectures

1171 **Corollary 2** (DeepSets, Transformers without positional enc.). Encoder architectures of either type
 1172 below satisfy Assumption 2 and Proposition 4:

- 1173 (a) *DeepSets / fully-connected GNN with global mean:* $T^{(\ell)}$ and $T^{(K+1)}$ are sample means;
 1174 (b) *Self-attention block with mean head:* $T^{(\ell)}$ are sample means; MLP_ℓ includes the
 1175 softmax-attention update.

1176 **Why max-pooling fails** The max functional $T_{\max}(x_{1:m}) = \max_i x_i$ is *not* Hadamard differentiable
 1177 at continuous laws: Its influence function is identically 0 whenever the maximum is attained at a
 1178 unique point and undefined when it is not. Consequently, the centered statistic $m^{1/2}\{T_{\max}(P_m) -$
 1179 $T_{\max}(P)\}$ has a *non-Gaussian* limit—the Gumbel extreme-value law—so Assumption 2(iii) fails.
 1180 Using max-pooling inside a deep encoder, therefore breaks the loss-CLT of Theorem 2. (Softmax
 1181 pooling with temperature $\tau > 0$, on the other hand, is smooth and becomes a valid AL pooler.)

1182 The table below summarises the status of common poolers.

	Pooler	AL / CLT?	Influence fcn. ψ_P in $L^2(P)$?
	Sample mean	✓	✓
1183	Huber M -estimator (δ fixed)	✓	✓
	Sample median	✓	✓
	Top- k or max	×	×
	Softmax ($\tau > 0$ fixed)	✓	✓

1184 **Generators** All neural generators considered in the experiments—MLPs, Transformer decoders,
 1185 and diffusion-score networks with fixed weights—are compositions of C^2 maps on finite-dimensional
 1186 spaces and therefore satisfy Assumption 3.

1187 **Smooth Approximation of Non-Regular Statistics** The theory developed here establishes that
 1188 Hadamard differentiability of the encoder ensure asymptotic normality and consistency and in
 1189 subsection 5.1 we develop the idea that our encoders learn sufficient statistics. But what if the
 1190 sufficient statistic of interest is not Hadamard differentiable? The sample maximum is a classic
 1191 example: it is the minimal sufficient statistic for the endpoint of a uniform distribution (see Example
 1192 3), yet it is not asymptotically normal.

1193 Let $X_1, \dots, X_n \sim \text{Uniform}(0, \theta)$. The sample maximum

$$X_{(n)} := \max\{X_1, \dots, X_n\}$$

1194 satisfies

$$n(\theta - X_{(n)}) \xrightarrow{d} \text{Exp}(1/\theta),$$

1195 so it converges to θ but its asymptotic distribution is exponential, not Gaussian. This occurs be-
 1196 cause the maximum is not a smooth functional of the empirical distribution: it fails Hadamard
 1197 differentiability, so the functional delta method does not apply.

1198 A natural remedy is to approximate the max by a smooth function. A standard choice is the *log-sum-*
1199 *exp*:

$$\text{LSE}_\lambda(X_1, \dots, X_n) = \frac{1}{\lambda} \log \left(\sum_{i=1}^n e^{\lambda X_i} \right).$$

1200 For fixed λ , this is Hadamard differentiable and thus amenable to the theory developed above. As
1201 $\lambda \rightarrow \infty$, $\text{LSE}_\lambda \rightarrow \max_i X_i$, so we recover the max in the limit.

1202 **Corollary 3** (Smooth approximation suffices for asymptotic normality). Let $T(P_m)$ be a non-smooth
1203 statistic (e.g., the maximum), and let $T^{(\lambda)}(P_m)$ be a family of smooth approximations (e.g., LSE_λ)
1204 such that $T^{(\lambda)}(P_m) \rightarrow T(P_m)$ pointwise. Then for any fixed λ , $T^{(\lambda)}(P_m)$ is Hadamard differentiable
1205 and admits asymptotically normal plug-in estimators. Moreover, if $\lambda_n \rightarrow \infty$ slowly as $n \rightarrow \infty$, this
1206 family can approximate $T(P_m)$ arbitrarily closely while retaining asymptotic normality.

1207 Thus, even when the true sufficient statistic is not regular, a Hadamard differentiable encoder can still
1208 be learned to approximate it. This ensures that the asymptotic guarantees from Theorem 2 continue
1209 to hold. This also highlight why we cannot use e.g. max-pooling in the encoder, since that would
1210 break our CLT.

1211 D.3 Embeddings and Predictive Sufficiency

1212 **Setting.** Let $\mathcal{M} \subset \mathcal{P}(\mathcal{X})$ be the statistical manifold introduced in Section D.2

1213 Here we assume the statistical manifold \mathcal{M} is d -dimensional (in the usual differential-geometric
1214 sense), so $\dim T_P \mathcal{M} = d$ for every $P \in \mathcal{M}$.

1215 For $P \in \mathcal{M}$ observe $S_m = (X_1, \dots, X_m) \stackrel{\text{i.i.d.}}{\sim} P$ and write the empirical measure $P_m =$
1216 $m^{-1} \sum_{i=1}^m \delta_{X_i}$.

1217 Throughout we use the *plug-in predictor* P_m . Given a statistic $T_m = \phi(P_m)$ with $\phi : \mathcal{M} \rightarrow \mathbb{R}^d$,
1218 define a measurable *reconstruction* map $R : \phi(\mathcal{M}) \rightarrow \mathcal{M}$ and set

$$P_m^\phi := R(T_m).$$

1219 **Definition 7** (Predictive sufficiency). The statistic $T_m = \phi(P_m)$ is *asymptotically predictive sufficient*
1220 if there exists a reconstruction R such that, for every $P \in \mathcal{M}$,

$$\|P_m - P_m^\phi\|_{\text{TV}} \xrightarrow[m \rightarrow \infty]{P^{\otimes m}} 0.$$

1221 This notion of sufficiency coincides with the one used in Section D.2: both ask that the predictor
1222 available to the decoder (here P_m^ϕ) converges in total variation to the full plug-in predictor P_m .

1223 **Theorem 3** (Embedding \iff Predictive sufficiency). Assume ϕ is C^1 and satisfies the encoder
1224 regularity conditions of Assumption 2. Then the following are equivalent.

1225 (i) Smooth embedding: ϕ is injective and its differential $d\phi_P : T_P \mathcal{M} \rightarrow \mathbb{R}^d$ is bijective for
1226 every $P \in \mathcal{M}$.

1227 (ii) Predictive sufficiency: $T_m = \phi(P_m)$ is asymptotically plug-in sufficient in the sense of
1228 Definition 7.

1229 *Proof.* Throughout, $\|\cdot\|_{\text{BL}}$ denotes the bounded-Lipschitz norm on signed measures, and $\|\cdot\|_{\text{TV}} \leq$
1230 $\|\cdot\|_{\text{BL}}$.

1231 **Step 1: (i) \implies (ii).**

1232 *Step 1(a): global inverse and Lipschitz constant.* Because ϕ is a C^1 diffeomorphism onto its image,
1233 the inverse-function theorem supplies, for every $P \in \mathcal{M}$, an open neighbourhood $U_P \subset \mathcal{M}$ on which
1234 $R \equiv \phi^{-1}$ is also C^1 . Shrink U_P so that the operator norm of dR_Q is bounded by some $L_P < \infty$ for
1235 all $Q \in U_P$; then R is L_P -Lipschitz on U_P under $\|\cdot\|_{\text{BL}}$.

1236 *Step 1(b): stochastic linearisation of $\phi(P_m)$.* Encoder regularity (Assumption 2) gives

$$\sqrt{m} \{ \phi(P_m) - \phi(P) \} = \frac{1}{\sqrt{m}} \sum_{i=1}^m \psi_P(X_i) + o_P(1) \quad \text{in } \mathbb{R}^d,$$

1237 so $\|\phi(P_m) - \phi(P)\| = O_P(m^{-1/2})$.

1238 *Step 1(c): reconstructing P_m .* For m large enough $P_m \in U_P$ with probability one, whence

$$\|P_m - R(\phi(P_m))\|_{\text{BL}} \leq L_P \|\phi(P_m) - \phi(P)\| = O_P(m^{-1/2}).$$

1239 Dividing by $\|\cdot\|_{\text{TV}}$ concludes plug-in sufficiency.

1240 **Step 2: (ii) \implies (i).**

1241 *Step 2(a): Continuity of ϕ .* Suppose $P_n \rightarrow P$ in \mathcal{M} but $\phi(P_n) \not\rightarrow \phi(P)$. Choose $\varepsilon > 0$ and a
 1242 subsequence (still indexed by n) with $\|\phi(P_n) - \phi(P)\| \geq \varepsilon$. For each n draw $S_m^{(n)} \sim P_n^{\otimes m_n}$ with
 1243 $m_n \uparrow \infty$ slowly enough that $\|P_{m_n}^{(n)} - P_n\|_{\text{TV}} \leq \varepsilon/4$ w.p. $\geq 1 - \varepsilon$. By sufficiency, $\|R(\phi(P_{m_n}^{(n)})) -$
 1244 $P_{m_n}^{(n)}\|_{\text{TV}} \leq \varepsilon/4$ with the same probability. The triangle inequality then forces $\|R(\phi(P_{m_n}^{(n)})) -$
 1245 $P\|_{\text{TV}} \geq \varepsilon/2$, contradicting $R(\phi(P_{m_n}^{(n)})) \xrightarrow{d} P$. Hence ϕ is continuous at every point.

1246 *Step 2(b): Injectivity of ϕ .* Assume $\phi(P_1) = \phi(P_2)$ with $P_1 \neq P_2$. Choose a measurable set B for
 1247 which $P_1(B) \neq P_2(B)$. Under $P_1^{\otimes m}$ we have $P_m(B) \rightarrow P_1(B)$ almost surely, while sufficiency
 1248 yields $R(T_m)(B) \rightarrow P_1(B)$. Repeating under $P_2^{\otimes m}$ forces $P_2(B) = P_1(B)$, contradiction. Hence
 1249 ϕ is injective.

1250 *Step 2(c): Injectivity of $d\phi_P$.* Suppose there is $v \in T_P\mathcal{M} \setminus \{0\}$ with $d\phi_P[v] = 0$. Pick a C^1 path
 1251 $t \mapsto P_t$ in \mathcal{M} with $P_0 = P$ and $\partial_t P_t|_0 = v$. Taylor expansion of $\phi(P_t)$ yields $\|\phi(P_t) - \phi(P)\| = o(t)$,
 1252 whereas $\|P_t - P\|_{\text{BL}} = \Theta(t)$. Setting $t = m^{-1/2}$ violates sufficiency exactly as in the previous step.
 1253 Hence $d\phi_P$ is injective; because the tangent and target spaces share the same (finite) dimension, it is
 1254 bijective.

1255 *Step 2(d): Smooth embedding.* Injectivity, continuity, and bijective differentials for all $P \in \mathcal{M}$ imply
 1256 that ϕ is a smooth embedding. \square

1257 **Remark** (Identifiability is automatic). *Because each $P \in \mathcal{M}$ already defines a unique predictive*
 1258 *distribution, any statistic that is plug-in sufficient must be injective; no separate identifiability*
 1259 *condition is required.*

E Extensions

E.1 Extension to Multiscale Settings

In many applications, data is naturally organized across multiple scales. For example, we may observe distributions of samples at a fine scale (e.g., single cells), grouped into entities at a coarser scale (e.g., patients), which themselves may belong to larger groups (e.g., hospitals). More generally, we may observe hierarchical data in which each level exhibits internal distributional structure.

Our framework naturally extends to such multiscale settings. At each scale s , we observe a set of units indexed by $i = 1, \dots, n^{(s)}$. Each unit i at scale s is associated with: a set of samples $S_{i,m}^{(s)} = \{x_{ij}^{(s)}\}_{j=1}^m$, drawn i.i.d. from a distribution $P_i^{(s)}$ and a higher-scale sample $x_i^{(s+1)} \in \mathcal{X}^{(s+1)}$, representing the corresponding entity at scale $s+1$.

The lower-scale distributions $P_i^{(s)}$ are drawn i.i.d. from a meta-distribution $Q^{(s)}$ over $\mathcal{P}(\mathcal{X}^{(s)})$, while the higher-scale samples $x_i^{(s+1)}$ are drawn from $P_i^{(s+1)}$, where $P_i^{(s+1)} \sim Q^{(s+1)}$.

Each lower-scale set $S_{i,m}^{(s)}$ defines an empirical measure

$$P_{i,m}^{(s)} = \frac{1}{m} \sum_{j=1}^m \delta_{x_{ij}^{(s)}} \in \mathcal{P}_m(\mathcal{X}^{(s)}).$$

At each scale we learn: an encoder $\mathcal{E}^{(s)} : \mathcal{P}_m(\mathcal{X}^{(s)}) \rightarrow \mathbb{R}^{d_s}$ mapping lower-scale empirical distributions into latent space, an encoder $\mathcal{E}^{(s+1)} : \mathcal{X}^{(s+1)} \rightarrow \mathbb{R}^{d_{s+1}}$ mapping higher-scale samples into the corresponding latent space, and generators $\mathcal{G}^{(s)} : \mathbb{R}^{d_s} \rightarrow \mathcal{P}(\mathcal{X}^{(s)})$ and $\mathcal{G}^{(s+1)} : \mathbb{R}^{d_{s+1}} \rightarrow \mathcal{P}(\mathcal{X}^{(s+1)})$ at each scale.

To link adjacent scales, we introduce deterministic maps

$$f^{(s)} : \mathbb{R}^{d_s} \rightarrow \mathbb{R}^{d_{s+1}} \quad \text{and} \quad g^{(s)} : \mathbb{R}^{d_{s+1}} \rightarrow \mathbb{R}^{d_s},$$

which project embeddings upward and downward between latent spaces.

We jointly train to enforce: *Approximate identity* at each scale:

$$\mathcal{G}^{(s)}(\mathcal{E}^{(s)}(S_{i,m}^{(s)})) \approx P_i^{(s)}, \quad \mathcal{G}^{(s+1)}(\mathcal{E}^{(s+1)}(x_i^{(s+1)})) \approx P_i^{(s+1)},$$

and *co-embedding consistency*: the mapped lower-scale embedding $f^{(s)}(\mathcal{E}^{(s)}(S_{i,m}^{(s)}))$ should align with the higher-scale embedding $\mathcal{E}^{(s+1)}(x_i^{(s+1)})$ and vice versa via $g^{(s)}$.

Formally, we optimize objectives of the form:

$$L = \mathfrak{d}(P_i^{(s)}, \mathcal{G}^{(s)}(\mathcal{E}^{(s)}(S_{i,m}^{(s)}))) \tag{4}$$

$$+ \mathfrak{d}(P_i^{(s+1)}, \mathcal{G}^{(s+1)}(\mathcal{E}^{(s+1)}(x_i^{(s+1)}))) \tag{5}$$

$$+ \|f^{(s)}(\mathcal{E}^{(s)}(S_{i,m}^{(s)})) - \mathcal{E}^{(s+1)}(x_i^{(s+1)})\|^2 \tag{6}$$

$$+ \|g^{(s)}(\mathcal{E}^{(s+1)}(S_{i,m}^{(s+1)})) - \mathcal{E}^{(s)}(x_i^{(s)})\|^2 \tag{7}$$

where \mathfrak{d} is a divergence or distance (e.g., KL divergence, Wasserstein distance) defined by the generative model. One natural approach would be to let $f^{(s)}, g^{(s)}$ both be the identity, forcing the model to learn a co-embedding across scales. But this may be too rigid and we might prefer more flexibility in practice.

This bi-directional coupling ensures that embeddings at adjacent scales are mutually predictive and geometrically aligned, while each scale individually satisfies distributional invariance and approximate identity. The framework naturally generalizes to hierarchies involving more than two scales by recursively composing the maps $f^{(s)}$ and $g^{(s)}$ across levels.

1291 **F Broader impacts**

1292 Generative distribution embeddings provide a general framework for modeling data across scales.
1293 They are broadly applicable to a wide variety of problems, including those with direct societal
1294 consequences, for example in healthcare. In these settings, it will be critical to consider any potential
1295 inequities induced by GDEs, as is the case for any modelling approach. Lastly, we acknowledge
1296 the environmental impact of this paper, which used nontrivial amounts of computational resources,
1297 estimated to be about 54kg CO₂.

References

- [1] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A Hilbert space embedding for distributions. In *Lecture Notes in Computer Science*, Lecture notes in computer science, pages 13–31. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 9783540752240,9783540752257. doi: 10.1007/978-3-540-75225-7_5.
- [2] Krikamol Muandet, K Fukumizu, Francesco Dinuzzo, and B Scholkopf. Learning from distributions via support measure machines. *Neural Information Processing Systems*, 25:10–18, 29 February 2012.
- [3] Junier B Oliva, B Póczos, and J Schneider. Distribution to Distribution Regression. *International Conference on Machine Learning*, 28(3):1049–1057, 16 June 2013.
- [4] Zoltan Szabo, Arthur Gretton, Barnabas Poczos, and Bharath Sriperumbudur. Two-stage sampled learning theory on distributions. In *Artificial Intelligence and Statistics*, pages 948–957. PMLR, 21 February 2015.
- [5] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017. ISSN 1935-8237,1935-8245. doi: 10.1561/22000000060.
- [6] Harrison Edwards and Amos Storkey. Towards a Neural Statistician. In *International Conference on Learning Representations*, 6 February 2017.
- [7] S I Amari and H Nagaoka. *Methods of information geometry*. 191, 2000.
- [8] K Carter, R Raich, W Finn, and A Hero. FINE: Fisher Information Nonparametric Embedding. *IEEE transactions on pattern analysis and machine intelligence*, 31(11):2093–2098, 14 February 2008. ISSN 0162-8828,1939-3539. doi: 10.1109/TPAMI.2009.67.
- [9] Yonghyeon Lee, Seungyeon Kim, Jinwon Choi, and F Park. A statistical manifold framework for point cloud data. *International Conference on Machine Learning*, pages 12378–12402, 2022.
- [10] Hannes Stärk, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, R Barzilay, and T Jaakkola. Dirichlet flow matching with applications to DNA sequence design. *International Conference on Machine Learning*, 235:46495–46513, 8 February 2024. doi: 10.48550/arXiv.2402.05841.
- [11] Oscar Davis, Samuel Kessler, Mircea Petrache, I Ceylan, Michael M Bronstein, and A Bose. Fisher flow matching for generative modeling over discrete data. *Neural Information Processing Systems*, abs/2405.14664:139054–139084, 23 May 2024. doi: 10.48550/arXiv.2405.14664.
- [12] Felix Otto. The geometry of dissipative evolution equations: the porous medium equation. 2001.
- [13] Doron Haviv, Aram-Alexandre Pooladian, Dana Pe’er, and Brandon Amos. Wasserstein flow matching: Generative modeling over families of distributions, 2024. URL <https://arxiv.org/abs/2411.00698>.
- [14] Lazar Atanackovic, Xi Zhang, Brandon Amos, Mathieu Blanchette, Leo J Lee, Yoshua Bengio, Alexander Tong, and Kirill Neklyudov. Meta Flow Matching: Integrating Vector Fields on the Wasserstein Manifold. In *The Thirteenth International Conference on Learning Representations*, 4 October 2024.
- [15] D Haviv, Russell Z Kunes, Thomas Dougherty, Cassandra Burdziak, T Nawy, Anna Gilbert, and D Pe’er. Wasserstein Wormhole: Scalable optimal transport distance with transformers. *International Conference on Machine Learning*, 235:17697–17718, 15 April 2024. doi: 10.48550/arXiv.2404.09411.
- [16] John W Fisher, III, A Ihler, and Paul A Viola. Learning informative statistics: A nonparametric approach. *Neural Information Processing Systems*, pages 900–906, 29 November 1999. doi: 10.5555/3009657.3009784.

- [17] Paul Joyce and Paul Marjoram. Approximately sufficient statistics and bayesian computation. *Statistical applications in genetics and molecular biology*, 7(1):Article26, 30 August 2008. ISSN 1544-6115,2194-6302. doi: 10.2202/1544-6115.1389.
- [18] Justin Alsing, Benjamin Wandelt, and Stephen Feeney. Massive optimal data compression and density estimation for scalable, likelihood-free inference in cosmology. *Monthly notices of the Royal Astronomical Society*, 477(3):2874–2885, 1 July 2018. ISSN 0035-8711,1365-2966. doi: 10.1093/mnras/sty819.
- [19] Yanzhi Chen, Dinghuai Zhang, Michael U Gutmann, Aaron Courville, and Zhanxing Zhu. Neural Approximate Sufficient Statistics for Implicit Models. In *International Conference on Learning Representations*, 2 October 2020.
- [20] Maxime Peyrard and Kyunghyun Cho. Meta-Statistical Learning: Supervised Learning of Statistical Inference, 2025.
- [21] M Zaheer, Satwik Kottur, Siamak Ravanbakhsh, B Póczos, R Salakhutdinov, and Alex Smola. Deep Sets. *Advances in neural information processing systems*, 30, 10 March 2017.
- [22] E Wagstaff, F Fuchs, Martin Engelcke, Michael A Osborne, and I Posner. Universal approximation of functions on sets. *Journal of machine learning research: JMLR*, 23(151):151:1–151:56, 5 July 2021. ISSN 1532-4435,1533-7928.
- [23] Lily H Zhang, Veronica Tozzo, J Higgins, and R Ranganath. Set norm and equivariant skip connections: Putting the deep in Deep Sets. *International Conference on Machine Learning*, 162:26559–26574, 23 June 2022. doi: 10.48550/arXiv.2206.11925.
- [24] Drew A Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K Lampinen, Andrew Jaegle, James L McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. SODA: Bottleneck diffusion models for representation learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23115–23127. IEEE, 16 June 2024. doi: 10.1109/cvpr52733.2024.02181.
- [25] Diederik P Kingma and M Welling. An Introduction to Variational Autoencoders. *Found. Trends Mach. Learn.*, 12(4):307–392, 6 June 2019. doi: 10.1561/22000000056.
- [26] Aude Genevay, Gabriel Peyre, and Marco Cuturi. Learning Generative Models with Sinkhorn Divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617. PMLR, 31 March 2018.
- [27] Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, R Badeau, and G Rohde. Generalized Sliced Wasserstein Distances. *Neural Information Processing Systems*, 32:261–272, 1 February 2019.
- [28] Jonathan Ho, Ajay Jain, and P Abbeel. Denoising Diffusion Probabilistic Models. *Neural Information Processing Systems*, abs/2006.11239:6840–6851, 19 June 2020.
- [29] Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. ProGen2: Exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978.e3, 15 November 2023. ISSN 2405-4712,2405-4720. doi: 10.1016/j.cels.2023.10.002.
- [30] Eric D Nguyen, Michael Poli, Marjan Faizi, A Thomas, Callum Birch-Sykes, Michael Wornow, Aman Patel, Clayton M Rabideau, Stefano Massaroli, Y Bengio, Stefano Ermon, S Baccus, and Christopher Ré. HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution. *Neural Information Processing Systems*, 36:43177–43201, 27 June 2023. doi: 10.48550/arXiv.2306.15794.
- [31] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 22 October 2020. ISSN 0001-0782,1557-7317. doi: 10.1145/3422622.
- [32] Y Lipman, Ricky T Q Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for generative modeling. *International Conference on Learning Representations*, abs/2210.02747, 6 October 2022.

- [33] José M Bernardo and Adrian FM Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009.
- [34] John Nash. The imbedding problem for riemannian manifolds. *Annals of mathematics*, 63(1): 20–63, 1956.
- [35] Julie Delon and Agnes Desolneux. A wasserstein-type distance in the space of gaussian mixture models. *SIAM Journal on Imaging Sciences*, 13(2):936–970, 2020.
- [36] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, 2, 2010.
- [37] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL <http://arxiv.org/abs/1708.07747>.
- [38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, June 2016. ISBN 9781467388511,9781467388528. doi: 10.1109/cvpr.2016.90.
- [39] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, December 2018. ISSN 1548-7091,1548-7105. doi: 10.1038/s41592-018-0229-2.
- [40] Caleb Weinreb, Alejo Rodriguez-Fraticelli, Fernando D Camargo, and Allon M Klein. Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science*, 367(6479), 14 February 2020. ISSN 0036-8075,1095-9203. doi: 10.1126/science.aaw3381.
- [41] Gokul Gowri, Xiao-Kang Lun, Allon M Klein, and Peng Yin. Approximating mutual information of high-dimensional variables using learned representations. In A Globerson, L Mackey, D Belgrave, A Fan, U Paquet, J Tomczak, and C Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 132843–132875. Curran Associates, Inc., 2024.
- [42] Xianghao Kong, Ollie Liu, Han Li, Dani Yogatama, and Greg Ver Steeg. Interpretable Diffusion via Information Decomposition. In *The Twelfth International Conference on Learning Representations*, 13 October 2023.
- [43] Omar O Abudayyeh and Jonathan S Gootenberg. Programmable biology through artificial intelligence: from nucleic acids to proteins to cells. *Nature Methods*, 21(8):1384–1386, 2024.
- [44] Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B Burkhardt, et al. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25): 7045–7063, 2024.
- [45] Julia Joung, Sai Ma, Tristan Tay, Kathryn R Geiger-Schuller, Paul C Kirchgatterer, Vanessa K Verdine, Baolin Guo, Mario A Arias-Garcia, William E Allen, Ankita Singh, et al. A transcription factor atlas of directed differentiation. *Cell*, 186(1):209–229, 2023.
- [46] Joseph M Replogle, Reuben A Saunders, Angela N Pogson, Jeffrey A Hussmann, Alexander Lenail, Alina Guna, Lauren Mascibroda, Eric J Wagner, Karen Adelman, Gila Lithwick-Yanai, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575, 2022.
- [47] Yiqun Chen and James Zou. Genept: a simple but effective foundation model for genes and cells built from chatgpt. *bioRxiv*, pages 2023–10, 2024.
- [48] Luke Funk, Kuan-Chung Su, Jimmy Ly, David Feldman, Avtar Singh, Britannia Moodie, Paul C Blainey, and Iain M Cheeseman. The phenotypic landscape of essential human genes. *Cell*, 185(24):4634–4653, 2022.

- [49] Netanel Loyfer, Judith Magenheimer, Ayelet Peretz, Gordon Cann, Joerg Bredno, Agnes Klochendler, Ilana Fox-Fisher, Sapir Shabi-Porat, Merav Hecht, Tsuria Pelet, Joshua Moss, Zeina Drawshy, Hamed Amini, Patriss Moradi, Sudharani Nagaraju, Dvora Bauman, David Shveiky, Shay Porat, Uri Dior, Gurion Rivkin, Omer Or, Nir Hirshoren, Einat Carmon, Alon Pikarsky, Abed Khalaileh, Gideon Zamir, Ronit Grinbaum, Machmud Abu Gazala, Ido Mizrahi, Noam Shussman, Amit Korach, Ori Wald, Uzi Izhar, Eldad Erez, Vladimir Yutkin, Yaacov Samet, Devorah Rotnemer Golinkin, Kirsty L Spalding, Henrik Druid, Peter Arner, A M James Shapiro, Markus Grompe, Alex Aravanis, Oliver Venn, Arash Jamshidi, Ruth Shemer, Yuval Dor, Benjamin Glaser, and Tommy Kaplan. A DNA methylation atlas of normal human cell types. *Nature*, 613(7943):355–364, 4 January 2023. ISSN 0028-0836,1476-4687. doi: 10.1038/s41586-022-05580-6.
- [50] Carl G de Boer, Eeshit Dhaval Vaishnav, Ronen Sadeh, Esteban Luis Abeyta, Nir Friedman, and Aviv Regev. Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. *Nature biotechnology*, 38(1):56–65, 2020.
- [51] Carl G De Boer and Timothy R Hughes. Yetfasco: a database of evaluated yeast transcription factor sequence specificities. *Nucleic acids research*, 40(D1):D169–D179, 2012.
- [52] Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, October 2018. ISSN 1548-7091,1548-7105. doi: 10.1038/s41592-018-0138-4.
- [53] Tristan Bepler and Bonnie Berger. Learning the protein language: Evolution, structure, and function. *Cell systems*, 12(6):654–669.e3, 16 June 2021. ISSN 2405-4712,2405-4720. doi: 10.1016/j.cels.2021.05.017.
- [54] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences of the United States of America*, 118(15), 13 April 2021. ISSN 0027-8424,1091-6490. doi: 10.1073/pnas.2016239118.
- [55] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan Dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 17 March 2023. ISSN 0036-8075,1095-9203. doi: 10.1126/science.ade2574.
- [56] Yuelong Shu and John McCauley. GISAID: Global initiative on sharing all influenza data - from vision to reality. *Euro surveillance : bulletin Europeen sur les maladies transmissibles [Euro surveillance : European communicable disease bulletin]*, 22(13), 30 March 2017. ISSN 1025-496X,1560-7917. doi: 10.2807/1560-7917.ES.2017.22.13.30494.
- [57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [59] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [60] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.
- [61] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

- 611 [62] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al.
612 Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- 613 [63] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information.
614 *Physical review. E, Statistical, nonlinear, and soft matter physics*, 69(6 Pt 2):066138, June 2004.
615 ISSN 1539-3755. doi: 10.1103/PhysRevE.69.066138.
- 616 [64] Netanel Loyfer, Jonathan Rosenski, and Tommy Kaplan. wgbstools: A computational suite for
617 DNA methylation sequencing data representation, visualization, and analysis. *bioRxiv*, page
618 2024.05.08.593132, 10 May 2024. doi: 10.1101/2024.05.08.593132.
- 619 [65] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the
620 monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.
- 621 [66] Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge
622 University Press, 2019.
- 623 [67] David Blackwell. An analog of the minimax theorem for vector payoffs. 1956.
- 624 [68] Aad W Van Der Vaart and Jon A Wellner. *Weak convergence*. Springer, 1996.

1298 NeurIPS Paper Checklist

1299 1. Claims

1300 Question: Do the main claims made in the abstract and introduction accurately reflect the
1301 paper's contributions and scope?

1302 Answer: [\[Yes\]](#)

1303 Justification: The abstract and introduction concretely state the main theoretical and empiri-
1304 cal results of the paper, and enumerate the demonstrated applications of our method.

1305 Guidelines:

- 1306 • The answer NA means that the abstract and introduction do not include the claims
1307 made in the paper.
- 1308 • The abstract and/or introduction should clearly state the claims made, including the
1309 contributions made in the paper and important assumptions and limitations. A No or
1310 NA answer to this question will not be perceived well by the reviewers.
- 1311 • The claims made should match theoretical and experimental results, and reflect how
1312 much the results can be expected to generalize to other settings.
- 1313 • It is fine to include aspirational goals as motivation as long as it is clear that these goals
1314 are not attained by the paper.

1315 2. Limitations

1316 Question: Does the paper discuss the limitations of the work performed by the authors?

1317 Answer: [\[Yes\]](#)

1318 Justification: We have a separate limitations subheading under the Discussion section. We
1319 clearly state key limitations of our method (assumption of exchangeability, etc).

1320 Guidelines:

- 1321 • The answer NA means that the paper has no limitation while the answer No means that
1322 the paper has limitations, but those are not discussed in the paper.
- 1323 • The authors are encouraged to create a separate "Limitations" section in their paper.
- 1324 • The paper should point out any strong assumptions and how robust the results are to
1325 violations of these assumptions (e.g., independence assumptions, noiseless settings,
1326 model well-specification, asymptotic approximations only holding locally). The authors
1327 should reflect on how these assumptions might be violated in practice and what the
1328 implications would be.
- 1329 • The authors should reflect on the scope of the claims made, e.g., if the approach was
1330 only tested on a few datasets or with a few runs. In general, empirical results often
1331 depend on implicit assumptions, which should be articulated.
- 1332 • The authors should reflect on the factors that influence the performance of the approach.
1333 For example, a facial recognition algorithm may perform poorly when image resolution
1334 is low or images are taken in low lighting. Or a speech-to-text system might not be
1335 used reliably to provide closed captions for online lectures because it fails to handle
1336 technical jargon.
- 1337 • The authors should discuss the computational efficiency of the proposed algorithms
1338 and how they scale with dataset size.
- 1339 • If applicable, the authors should discuss possible limitations of their approach to
1340 address problems of privacy and fairness.
- 1341 • While the authors might fear that complete honesty about limitations might be used by
1342 reviewers as grounds for rejection, a worse outcome might be that reviewers discover
1343 limitations that aren't acknowledged in the paper. The authors should use their best
1344 judgment and recognize that individual actions in favor of transparency play an impor-
1345 tant role in developing norms that preserve the integrity of the community. Reviewers
1346 will be specifically instructed to not penalize honesty concerning limitations.

1347 3. Theory assumptions and proofs

1348 Question: For each theoretical result, does the paper provide the full set of assumptions and
1349 a complete (and correct) proof?

Answer: [Yes]

Justification: Theoretical results are stated with complete proof and assumptions in the Appendix of the paper. Informal versions of theoretical results are provided in the main text.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All experimental details are provided in the Appendix. Moreover, all experimental results can be reproduced by running the code in the provided (anonymized Github repository). Models can be trained using the appropriate experiment configs in the `config/experiment/` directory, and figures from the paper can be reproduced by running notebooks in the `notebooks/` directory.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

1404 some way (e.g., to registered users), but it should be possible for other researchers
1405 to have some path to reproducing or verifying the results.

1406 5. Open access to data and code

1407 Question: Does the paper provide open access to the data and code, with sufficient instruc-
1408 tions to faithfully reproduce the main experimental results, as described in supplemental
1409 material?

1410 Answer: [Yes]

1411 Justification: Code and datasets are made publicly available, and code necessary to reproduce
1412 results are provided with documentation.

1413 Guidelines:

- 1414 • The answer NA means that paper does not include experiments requiring code.
- 1415 • Please see the NeurIPS code and data submission guidelines ([https://nips.cc/](https://nips.cc/public/guides/CodeSubmissionPolicy)
1416 [public/guides/CodeSubmissionPolicy](https://nips.cc/public/guides/CodeSubmissionPolicy)) for more details.
- 1417 • While we encourage the release of code and data, we understand that this might not be
1418 possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not
1419 including code, unless this is central to the contribution (e.g., for a new open-source
1420 benchmark).
- 1421 • The instructions should contain the exact command and environment needed to run to
1422 reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- 1423 • The authors should provide instructions on data access and preparation, including how
1424 to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- 1425 • The authors should provide scripts to reproduce all experimental results for the new
1426 proposed method and baselines. If only a subset of experiments are reproducible, they
1427 should state which ones are omitted from the script and why.
- 1428 • At submission time, to preserve anonymity, the authors should release anonymized
1429 versions (if applicable).
- 1430 • Providing as much information as possible in supplemental material (appended to the
1431 paper) is recommended, but including URLs to data and code is permitted.

1433 6. Experimental setting/details

1434 Question: Does the paper specify all the training and test details (e.g., data splits, hyper-
1435 parameters, how they were chosen, type of optimizer, etc.) necessary to understand the
1436 results?

1437 Answer: [Yes]

1438 Justification: All experimental details (including train/test splits and model implementation
1439 choices) are provided in the Appendix, and can be found in the accompanying (anonymized)
1440 Github repository.

1441 Guidelines:

- 1442 • The answer NA means that the paper does not include experiments.
- 1443 • The experimental setting should be presented in the core of the paper to a level of detail
1444 that is necessary to appreciate the results and make sense of them.
- 1445 • The full details can be provided either with the code, in appendix, or as supplemental
1446 material.

1447 7. Experiment statistical significance

1448 Question: Does the paper report error bars suitably and correctly defined or other appropriate
1449 information about the statistical significance of the experiments?

1450 Answer: [Yes]

1451 Justification: Standard errors are reported where relevant and feasible.

1452 Guidelines:

- 1453 • The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The appendix includes details of the compute resources used for this work. Full internal cluster details will be released after the double-blind period ends.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We adhere to the code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss potential societal impacts in the broader impacts section in Appendix of the paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not release datasets or models with high risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the datasets, code, and models used in the paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The code provided in the accompanying repository are well-documented.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: We do not perform crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- 1609 • Depending on the country in which research is conducted, IRB approval (or equivalent)
1610 may be required for any human subjects research. If you obtained IRB approval, you
1611 should clearly state this in the paper.
- 1612 • We recognize that the procedures for this may vary significantly between institutions
1613 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
1614 guidelines for their institution.
- 1615 • For initial submissions, do not include any information that would break anonymity (if
1616 applicable), such as the institution conducting the review.

1617 16. Declaration of LLM usage

1618 Question: Does the paper describe the usage of LLMs if it is an important, original, or
1619 non-standard component of the core methods in this research? Note that if the LLM is used
1620 only for writing, editing, or formatting purposes and does not impact the core methodology,
1621 scientific rigorousness, or originality of the research, declaration is not required.

1622 Answer: [NA]

1623 Justification: LLMs are not an important component of this work.

1624 Guidelines:

- 1625 • The answer NA means that the core method development in this research does not
1626 involve LLMs as any important, original, or non-standard components.
- 1627 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)
1628 for what should or should not be described.