Categorical Generative Model Evaluation via Synthetic Distribution Coarsening

Florence Regol **Mark Coates**
McGill University, International Laboratory on Learning Systems (ILLS) and Mila - Quebec AI Institute

Abstract

As we expect to see a rapid integration of generative models in our day to day lives, the development of rigorous methods of evaluation and analysis for generative models has never been more pressing. Multiple works have highlighted the shortcomings of widely used metrics and exposed how they fail to behave as expected in some settings. So far, the response has been to use a variety of metrics that target different desirable and interpretable properties such as fidelity, diversity, and authenticity, to obtain a clearer picture of a generative model's capabilities. These methods mainly focus on ordinal data and they all suffer from the same unavoidable issues stemming from estimating quantities of high-dimensional data from a limited number of samples. We propose to take an alternative approach and to return to the synthetic data setting where the ground truth is known. We focus on nominal categorical data and introduce an evaluation method that can scale to the high-dimensional settings often encountered in practice. Our method involves successively binning the large space to obtain smaller probability spaces and coarser distributions where meaningful statistical estimates can be obtained. This allows us to provide probabilistic guarantees and sample complexities and we illustrate how our method can be applied to distinguish between the capabilities of several state-of-the-art categorical models.

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

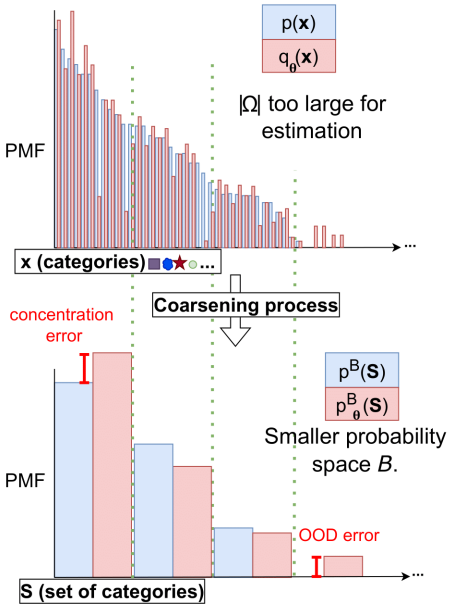


Figure 1: Overview of our coarsening evaluation method. We move from a prohibitively large probability space Ω to a coarser probability space $\mathcal{B}$, where we can evaluate quantities of interest with statistical significance.

1 INTRODUCTION

We are now able to generate realistic text (OpenAI, 2023), images (Rombach et al., 2022), audio speech (Shibuya et al., 2023) and videos (Yu et al., 2023). In other words, given a set of samples $x \sim p(x)$ associated with a very large probability space Ω, learning a distribution q_θ that can generate "believable samples" has largely been achieved, where "believable samples" refer to samples $\hat{x}$ where $p(\hat{x})$ is high.

While impressive, this does not imply that the more advanced task of learning the distribution, i.e., $q_\theta = p$, has been solved. In fact, important issues indicative of this shortcoming persist in the literature. Seemingly well-performing generative models can fail to properly handle Out-Of-Distribution (OOD) samples (Nalisnick

et al., 2019; Yang et al., 2022; Wang et al., 2023), and using data generated by those models is considered problematic (Veselovsky et al., 2023; Shumailov et al., 2023). Generative models and neural networks in general tend to be overconfident in their prediction which can contribute to model collapse in data generation, as emphasized by Shumailov et al. (2023). These issues imply that $q_\theta \neq p$.

This shortcoming has mainly been attributed to the inadequacy of our evaluation framework for generative models (Alaa et al., 2022; Naeem et al., 2020). When accessible, the most widely used metric is the empirical negative log likelihood (NLL). In practice, even for models where it is considered "available", it is often an approximation based on upper bounds. Although principled, the reliance on the NLL of held out test data has been associated with performance assessment issues (Theis et al., 2016; Jiralerspong et al., 2023). A simple example from (van den Oord and Schrauwen, 2014) shows how low NLL models can still generate poor samples – a model based on the ground truth but nearly completely corrupted with noise can still have a high likelihood. Beyond this, sample-based estimators of log likelihood metrics can fail to detect a model that incorrectly identifies low probability regions (Nalisnick et al., 2019). It is also challenging to interpret lower bounds of the log-likelihood and determine how well a model is actually approximating the distribution. As a result, there have been persistent concerns about the quality and utility of evaluation metrics (Theis et al., 2016; Novikova et al., 2017; Wu et al., 2017; Sajjadi et al., 2018; Borji, 2019; Garbacea et al., 2019; Zhou et al., 2019; Caccia et al., 2020; Celikyilmaz et al., 2020; Nagarajan et al., 2021; Thompson et al., 2022), and developing expressive and robust evaluation metrics is an active area of research (Alaa et al., 2022; Naeem et al., 2020; Khayatkhoei and AbdAlmageed, 2023; Jiralerspong et al., 2023).

While such efforts are moving in the right direction, the sampling approximation error is too great to be overcome if our goal is to meaningfully assess whether $p = q_\theta$. The latest advancements in evaluation methods that are targeting this goal (Alaa et al., 2022; Naeem et al., 2020; Khayatkhoei and AbdAlmageed, 2023) all heavily rely on having a meaningful distance metric for the probability space, which is not available for general categorical data. Moreover, high dimensionality is still the main challenge that impedes reliable application of these methods (Jiralerspong et al., 2023).

Consequently, we argue that for proper assessment, we need to consider the synthetic setting where the ground truth distribution p is known. Being able to ensure that a model is behaving as expected in a controlled environment is a crucial part of model development. We believe that the current lack of robust synthetic evaluation is actively harming the development of research in generative modeling. We emphasize that even for this easier task, with known ground truth distribution, for a probability space Ω, the provably optimal sample complexity (number of samples m) to distinguish whether two distributions are either equal or ϵ-far in total variation distance d_{tv} is of order $m = \Theta\big(\sqrt{|\Omega|}\big)$ (Diakonikolas et al., 2021). The main issue is not that the ground truth distribution is unknown; it is actually the high-dimensionality. This motivates our proposal for a categorical generative model evaluation framework.

At a high level, our solution involves coarsening the impractical Ω and performing statistical tests on smaller induced probability spaces. A preliminary approach towards the development of this work was described in Regol et al. (2023). See Figure 1 for a high level view of our proposal. This approach is sensible; if a model is unable to assign the correct probability mass to a set of elements of a coarsened space, then it is *necessarily* also unable to assign the correct probability mass to any partitioning of this set, including to individual elements (which corresponds to the initial probability space). Moreover, this approach provides an interpretable evaluation procedure with statistical guarantees. The framework allows us to assess how well a generative model manages to avoid generating out-of-distribution samples (i.e., samples where the true distribution is zero). The framework also allows us to detect when a generative model is overly concentrated, i.e., it generates too many samples for elements that have a high true probability. We can also gain better insights into which types of distributions a particular generative model can fit. We propose a procedure for constructing a ground truth distribution that retains key real-world dataset characteristics and illustrate how this synthetic distribution can be used to meaningfully rank four recent categorical generative models.

Our key contribution is the development of a novel procedure for evaluating (multivariate) categorical generative models on synthetic data that 1) is designed for the realistic large sample space setting (on the order of billions of elements); 2) provides statistical guarantees; 3) is robust to different patterns of mismatch between p, q; and 4) offers interpretable results.

2 RELATED WORK

Our work is related to recently proposed metrics for evaluating generative models and also to distribution testing. Aside from NLL, there are more task-oriented methods of evaluation based on generated samples (Inception Score (IS) (Salimans et al., 2016) and Fréchet Inception Distance (FID) (Heusel et al., 2017) metrics,

for example). While effective, such task-focused assessment procedures are not equipped to distinguish between different types of failure (Sajjadi et al., 2018) and are inherently tied to the task at hand. **Coverage metrics:** Recent work has proposed metrics to assess the mismatch between distributions (Sajjadi et al., 2018; Kynkäänniemi et al., 2019; Naeem et al., 2020; Djolonga et al., 2020; Liu et al., 2021; Alaa et al., 2022) through the interpretable notions of precision and recall, but the methods are only applicable to the continuous or ordered discrete domain, because they rely on access to a meaningful distance metric in the distribution space. This interpretation has also been called into question by recent works (Khayatkhoei and AbdAlmageed, 2023; Jiralerspong et al., 2023). Please see Appendix 9 for a more detailed discussion of related work.

Distribution testing. The problem of assessing how close a discrete distribution q is to some reference distribution p is well studied and is known as the *identity testing* problem (Batu et al., 2001; Diakonikolas et al., 2021; Canonne, 2020b, 2022). Most work on this topic aims to provide algorithm existence results and to develop complexity bounds on the number of samples required to estimate quantities of interest, such as the total variation distance. See (Canonne, 2020b) for a review. The focus of this line of research is not empirical, so proposed algorithms often involve constants that cannot be evaluated, which makes it challenging to apply them in practice.

3 PROBLEM SETTING

We consider discrete distributions $q_1, q_2, \dots, q_N$ and p defined over a nominal categorical probability space Ω. Our goal is to assess which generative model q_i is the closest to the ground truth distribution p in total variation: $d_{tv}(p,q) = \frac{1}{2}||\mathbf{p}-\mathbf{q}||_1$, where $\mathbf{p}$ denotes the vectorized probability mass function (pmf) values of p (with some arbitrary but consistent ordering). In our setting, since we define a synthetic dataset, we have complete access to and control over p. So that we can apply our procedure to all generative models, we assume that we only have sample access to q. The size of the sample space Ω is assumed to be prohibitively large for standard statistical tests in the original space.

We also require that the evaluation framework provides interpretable results. Hence, from the results of our procedure, we should be able to answer the following questions tied to common failures of modern generative models:

Does the generative model generate OOD samples? Given a ground truth distribution p, we can define the OOD set as $\Omega^o = \{x;\, p(x) = 0\}$. Hence, we want a metric that can assess $q_\theta(\Omega^o)$.

Is the generative model over-concentrated? Given a ground truth distribution p and a specified threshold η , we can define a "likely" set as $\Omega^{likely} = \{x;\, p(x) \geq \eta\}$. Hence, we want a metric that can assess how far $q_\theta(\Omega^{likely})$ is from $p(\Omega^{likely})$.

4 METHODOLOGY

Preliminaries We denote all partitions of a set Ω by $\rho(\Omega) = \{\{\mathcal{A}_1, \dots\} | \cup_i \mathcal{A}_i = \Omega, |\mathcal{A}_i| > 0, \mathcal{A}_j \cap \mathcal{A}_i = \emptyset \, \forall i \neq j\}$ and all partitions of size k by $\rho^k(\Omega) = \{\mathcal{B} \in \rho(\Omega), |\mathcal{B}| = k\}$. Given a probability space Ω , we can construct a new smaller probability space from a partitioning $\mathcal{B}^k \in \rho^k(\Omega)$ provided that $k < |\Omega|$. Any distribution p associated with the initial space Ω will then induce a binned distribution $p^{\mathcal{B}^k}$ on this new space: $p^{\mathcal{B}^k}_{\mathcal{A}_i} \triangleq \sum_{x \in \mathcal{A}_i} p_x$. Consequently, we can readily see by the triangle inequality that the total variation error between two distributions p, q on space Ω is always equal to or greater than the error on a partitioned space, i.e.,

$$\mathcal{B} \in \rho(\Omega) \implies d_{tv}(p^{\mathcal{B}}, q^{\mathcal{B}}) \leq d_{tv}(p,q). \tag{1}$$

Therefore, if we build a sequence of probability spaces $\{\mathcal{B}^1, \mathcal{B}^2 \dots, \mathcal{B}^K\}$ with the initial space $\mathcal{B}^K \in \rho(\Omega)$ as the end point and construct each preceding space as a partitioning of the previous ($\mathcal{B}^{i-1} \in \rho(\mathcal{B}^i)$), we will have that the d_{tv} error and the granularity ($|\mathcal{B}|$) will be increasing over the sequence, i.e., $|\mathcal{B}^1| \leq \dots \leq |\mathcal{B}^{i-1}| \leq |\mathcal{B}^i| \leq \dots \leq |\Omega|$, and:

$$d_{tv}(p^{\mathcal{B}^{i-1}}, q^{\mathcal{B}^{i-1}}) \leq d_{tv}(p^{\mathcal{B}^i}, q^{\mathcal{B}^i}) \dots \leq d_{tv}(p,q). \tag{2}$$

4.1 Proposal

Our proposal can be summarized as testing a generative model at increasingly fine granularities by considering the binned distributions over a sequence of probability spaces $\{\mathcal{B}^1, \mathcal{B}^2 \dots\}$. If a model q_1 is close enough to p over $\mathcal{B}^i$: $d_{tv}(p^{\mathcal{B}^i}, q_1^{\mathcal{B}^i}) \leq \epsilon_{test}$, but model q_2 is not: $d_{tv}(p^{\mathcal{B}^i}, q_2^{\mathcal{B}^i}) \geq \epsilon_{test}$, this allows us to state that, at granularity $|\mathcal{B}^i|$;

$$d_{tv}(p^{\mathcal{B}^i}, q_1^{\mathcal{B}^i}) \leq d_{tv}(p^{\mathcal{B}^i}, q_2^{\mathcal{B}^i}). \tag{3}$$

It is important to note that this result *does not* allow us make any statement with respect to the initial space. For example, it does not allow us to state definitively that $d_{tv}(p, q_1) \leq d_{tv}(p, q_2)$.

Limitation. This is an important limitation of our work. We cannot guarantee that the ordering of the generative models in the original probability space will not be reversed in the induced probability space, i.e., $d_{tv}(p, q_1) \leq d_{tv}(p, q_2) \implies d_{tv}(p^{\mathcal{B}}, q_1^{\mathcal{B}}) \leq d_{tv}(p^{\mathcal{B}}, q_2^{\mathcal{B}})$. We analyze the necessary conditions for this to occur in more details in Appendix 7.

In summary, this issue arises if a partitioning $\mathcal{B}$ ends up grouping together some elements that are highly overestimated by q_1, and others highly underestimated, which is then being "hidden" by the averaging. It is possible that this occurs to a larger extent than by q_2 and enough to erase the initial performance gap. We reduce the likelihood of this happening by randomly generating multiple $\mathcal{B}$ that groups different elements together, but we cannot eliminate this effect.

4.1.1 Choosing the bins

So far, the requirements for the sequence $\{\mathcal{B}^1, \mathcal{B}^2 \dots, \mathcal{B}^K\}$ are $\mathcal{B}^K \in \rho(\Omega)$ and $\mathcal{B}^{i-1} \in \rho(\mathcal{B}^i)$. For our purposes, we additionally want $d_{tv}(p^{\mathcal{B}}, q^{\mathcal{B}})$ to be as close as possible to $d_{tv}(p, q)$. The gap $\delta = d_{tv}(p, q) - d_{tv}(p^{\mathcal{B}}, q^{\mathcal{B}})$ should be minimized. This cannot be done directly as q is unknown; we can however take advantage of our control over p to minimize the component of the error associated with p. This can be achieved by basing the construction of the sequence on a "near-Δ" partitioning of the probability space of p, as defined below.

Given a distribution p and a tolerance parameter $\Delta \in [0, 1)$, we define a near-Δ set Ω_p^{Δ} as a partitioning of Ω ($\Omega_p^{\Delta} \in \rho(\Omega)$) of minimum cardinality that contains sets of elements in which the maximum difference between the probability mass of any two elements is at most Δ, i.e.:

$$\Omega_p^{\Delta} = \underset{\mathcal{X} \in \rho_p^{\Delta}(\Omega)}{\arg\min} |\mathcal{X}|, \tag{4}$$

$$\rho_p^{\Delta}(\Omega) = \{\{\mathcal{S}_1^{\Delta}, \dots\} \, s.t. \, |p_{x_j} - p_{x_k}| \leq \Delta \quad \forall x_j, x_k \in \mathcal{S}_i^{\Delta}\} \tag{5}$$

Given a fixed Δ and a distribution p, we can use the following procedure to find a partitioning Ω_p^{Δ} . We commence with an element with maximum probability mass $p_{\max}$. We then construct the first set of the partition ($\mathcal{S}_1^{\Delta}$) by including all elements x such that $p_x \geq p_{\max} - \Delta$. After removing all such elements, we repeat the process, denoting by $p_{\max}$ the largest probability mass for the remaining elements, and constructing the second set of the partitioning. We iterate until all elements have been included in a set. We provide the algorithm in the Appendix 3, together with a short proof that this procedure produces a partitioning with the smallest cardinality. For a general p, this algorithm requires $O(\Omega)$ operations. Since we have control over the design of p, we avoid this by operating in the reverse fashion: (i) specify a target near-Δ partitioning; and then (ii) assign probability mass to the elements in the sets of the partitions, ensuring that the Δ-constraint is satisfied.

We start to build the sequence by setting the first binned probability space $\mathcal{B}^1$ to the constructed near-Δ partitioning, $\mathcal{B}^1 \triangleq \Omega_p^{\Delta}$. The benefit of this is that we have bounded the component of the error $\delta = d_{tv}(p, q) - d_{tv}(p^{\mathcal{B}}, q^{\mathcal{B}})$ that is induced by averaging p — it can be at most $|\Omega_p^{\Delta}|\Delta/2$.

We then iteratively generate the next probability space $\mathcal{B}^{i+1}$ by choosing at random any set that contains more than one element, $\mathcal{A} \sim U(\{\mathcal{A} \in \mathcal{B}^i; |\mathcal{A}| > 1\})$ and splitting it in half randomly. By following this process we ensure that the requirements $\mathcal{B}^K = \Omega$ and $\mathcal{B}^{i-1} \in \rho(\mathcal{B}^i)$ are respected. Moreover, the cardinalities of the induced probability spaces that we consider are equal to s plus some constant: $|\mathcal{B}^i| = |\Omega_p^{\Delta}| + i - 1$, and are thus dramatically smaller than $|\Omega|$. We provide the details of the sequence construction $\{\mathcal{B}^1, \mathcal{B}^2, \dots\}$ as a component of the overall evaluation procedure in Algorithm 1.

4.1.2 Error Estimates

Now that we have described how the sequence of partitions is derived, we can specify how to obtain estimates of the error at each probability space granularity. These estimates allow us to rank the models using the d_{tv} estimates. A straightforward estimator of the total variation is the empirical "learning estimator" Canonne (2020b). Although not optimal Valiant and Valiant (2017); Diakonikolas et al. (2021), this method has the advantage of offering an explicit expression for the number of samples m required to provide probabilistic bounds. Given a set of m samples from q ($\{\tilde{x}_i\}_{i=1}^m \tilde{x}_i \sim q$), the empirical total variation estimator $\hat{T}^m$ is:

$$\hat{T}^m \triangleq d_{tv}(p, \tilde{q}) = \frac{1}{2} \sum_{x \in \Omega} |p_x - \tilde{q}_x|, \tag{6}$$

$$\text{where} \quad \tilde{q}_x = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}[\tilde{x}_i = x] \tag{7}$$

The following theorem, which builds directly on a result from Canonne (2020a), provides the number of samples required to state that with some probability, the true total variation is within a range centred at this $\hat{T}^m$ estimator. We provide a simple proof in Appendix 2.

Theorem 4.1. *Given a discrete distribution p with associated sample space $\mathcal{B}$, m samples from a distribution q with the same sample space $\mathcal{B}$, and an error tolerance*

$\epsilon_{test} \in (0, 1]$, *provided that:*

$$\epsilon_{test} \geq \max(\sqrt{\frac{|\mathcal{B}|}{m}}, \sqrt{\frac{2\ln(2/\delta)}{m}}). \tag{8}$$

we can be at least $1-\delta$ *confident that the true total variation* $d_{tv}(p, q)$ *is within the interval* $[\hat{T}^m - \epsilon_{test}, \hat{T}^m + \epsilon_{test}]$.

The next theorem follows from Theorem 4.1, and provides a basis to compare two generative models:

Theorem 4.2. *Given a discrete distribution* p *with associated sample space* $\mathcal{B}$*, and* m *samples from the distributions* q *and* q' *with the same sample space* $\mathcal{B}$*, denote by* $\hat{T}^m_q$ *and* $\hat{T}^m_{q'}$ *the empirical total variation estimators of* q *and* q'*, respectively. For an error tolerance* $\epsilon_{test} \in (0, 1]$ *s.t.* (32) *holds for a selected constant* $\delta \in (0, 1)$*, the random quantity* $\hat{T}^m_q - \hat{T}^m_{q'}$ *will fall within the following interval:*

$$\hat{T}^m_q - \hat{T}^m_{q'} \in [d_{tv}(p, q) - d_{tv}(p, q') \pm 2\epsilon_{\text{test}}] \tag{9}$$

with at least $(1-\delta)^2$ *probability.*

Appendix 5 contains the proof. We apply this result as follows in our procedure: if $\hat{T}^m_q - \hat{T}^m_{q'} - 2\epsilon_{\text{test}} > 0$, then with at least $(1-\delta)^2$ confidence we have $d_{tv}(p, q) > d_{tv}(p, q')$, indicating that model q' is better than model q.

4.1.3 Evaluation procedure

A complete description of the evaluation procedure is provided in Algorithm 1. Recalling Eqn. 2, our procedure generates a sequence of pairs of induced distributions $(p^{\mathcal{B}^i}, q^{\mathcal{B}^i})$ with associated sample spaces $\mathcal{B}^i$, with the guarantee that $\epsilon^{p^{\mathcal{B}^i}, q^{\mathcal{B}^i}}$ approaches the true total variation. On the one hand, it is desirable to compare the distributions at the finest granularity. However, for a fixed number of samples, as the granularity becomes finer, the accuracy of the d_{tv} estimates decreases. This is captured by Theorem 4.1, and can be visualized in Figure 2. The figure shows the lowest error ϵ_{test} that meets the requirements in (32) to apply Theorem 4.1 for different probability spaces to achieve an acceptably low significance level for a reasonable number of samples (100,000).

Metric B^***:** As a result, a good trade-off can be obtained by comparing the generative models based on the total variation error at the granularity $\mathcal{B}$ that is at the breaking point before the minimum error threshold starts increasing, i.e., at $|\mathcal{B}^*| = \lceil 2\ln(2/\delta) \rceil$. This provides a simple and efficient way to compare generative models with an interpretable metric.

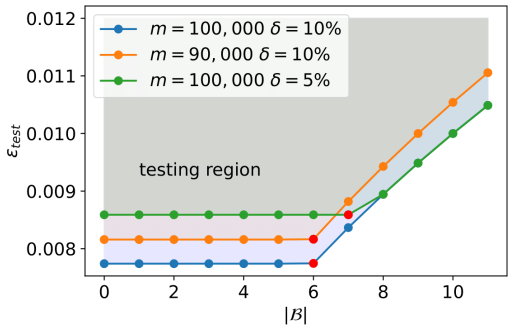


Figure 2: Error threshold that we can obtain over varying size of sample space, for 3 different cases with varying number of samples and probability significance δ.

Slope: Lastly, as we will show through empirical results, the rate at which the estimated total variation increases across different granularities can provide insights into the underlying behavior of the distribution w.r.t. the reference distribution p. As a result, in addition to reporting the individual $B^1, \ldots,$ we fit a linear regression on the estimates and report the slope. In the region where there are enough samples to provide a reasonable estimate of d_{tv}, i.e., relatively small $|B^i|$, then if the slope is close to zero, it indicates that the error is evenly distributed across the space. If the slope is increasing, then the error is concentrated in some sets of the partitioning.

Out-of-distribution samples: Our total variation-based metric of coarser distributions leads to a natural way to assess how much mass a generative model assigns to the out-of-distribution set Ω^o. By construction, for a sufficiently small Δ, our coarser space $\mathcal{B}^1 = \Omega^{\Delta}_p$ will have Ω^o as an element (from Eqn. 5, since $p_x = 0 \quad \forall x \in \Omega^o$, we will have $\mathcal{S}^{\Delta} = \Omega^o$). Consequently, all subsequent granular spaces $\{\mathcal{B}^1, \ldots, \}$, including $\mathcal{B}^*$, will have a set of elements that are a partitioning of Ω^o. These can be combined together to obtain the total variation of the set.

Hence we can compute and report:

$$OOD \triangleq |p(\Omega^o) - q_\theta(\Omega^o)| = q_\theta.(\Omega^o) \tag{10}$$

We can therefore directly assess $q_\theta(\Omega^o)$, as per our stated requirement.

Over-concentration: Next, to assess whether a model is over-concentrated on high probability elements using our method, we can focus on the highest probability element of $p^{\mathcal{B}}$. We choose to focus on our main granularity of reference, $\mathcal{B}^*$. This will define our

likely set Ω^{likely} from a threshold $\eta = \max(\mathbf{p}^{\mathcal{B}^*})$. We then report the difference:

$$conc \triangleq q_\theta(\Omega^{likely}) - p(\Omega^{likely}). \tag{11}$$

If this value is positive, it indicates that the generative model is over-concentrating probability mass on high probability elements.

Algorithm 1 Complete procedure with time and memory complexity denoted by $\mathbf{\Omega}(), o()$ symbols.

Input: Target distribution p, sample access to distribution q, test error ϵ_{test}, significance prob. δ, tolerance Δ, number of samples m, num random partitions $t = 50$.
Construct the near$-\Delta$ regions Ω_p^Δ using algorithm Appendix 3.1. Time: $\mathbf{\Omega}(1)$, Space: $\mathbf{\Omega}(1)$
Obtain m samples from q. Time: $o(m)$, Space: $o(m)$
Set $\mathcal{B}^1 = \Omega_p^\Delta$.
Compute the maximum valid granularity for ϵ_{test}: $|\mathcal{B}^{max}| = \lfloor \epsilon_{test}^2 m \rfloor$
$T = |\mathcal{B}^{max}| - |\mathcal{B}^1|$
for $i \in [1, \dots, T]$ **do**
  Granularity $g = |\mathcal{B}^i|$
  for $[1, \dots, t]$ **do**
    Compute $p^{\mathcal{B}^i}$.Time: $\mathbf{\Omega}(g)$, Space: $\mathbf{\Omega}(g)$
    Compute the estimate $\hat{T}_g^m$ from Eqn (7) Time: $\mathbf{\Omega}(m)$, Space: $\mathbf{\Omega}(1)$
    Select a random set $\mathcal{A}_i \in \mathcal{B}^k$
    Evenly split $\mathcal{A}_i$ by random in $\mathcal{A}_i', \mathcal{A}_i''$ Time: $\mathbf{\Omega}(m)$, Space: $\mathbf{\Omega}(1)$
    Construct the new probability space $\mathcal{B}^{k+1} = \mathcal{B}^k \setminus \mathcal{A}_i \cup \{\mathcal{A}_i' \cup \mathcal{A}_i''\}$
  end for
  Compute the average over the t estimates $\hat{T}_g^m$ and store it.
end for
Return all $\hat{T}_g^m$ Total time complexity: $\mathbf{\Omega}(mtT(|\Omega_p^\Delta| + T))$, Total space complexity: $\mathbf{\Omega}(|\Omega_p^\Delta| + T)$

4.2 Designing synthetic test distributions

We base the construction of the synthetic target distribution p on key characteristics shared across different real-data distributions. The goal is not to reproduce a pmf as sophisticated as real data, but to construct a task that could be viewed as a necessary first test that a model should be able to address.

First, to be representative of real datasets which are the target of deep learning generative models, the dimensionality of the space considered should be very high. We consider $|\Omega|$ in the order of billions. Second, we assume that the positive space $\Omega^+ \triangleq \{x; p_x > 0\}$ is also very large, but small relative to the whole space $|\Omega^+|/|\Omega| \approx 0$. This assumption holds for almost all practical applications of generative models. For example, if we sequentially sample letters randomly, the chance of forming a coherent sentence is next to zero.

Recalling our construction of $\mathcal{B}^1$, we see that the cardinality $|\mathcal{B}^1|$ is driven by $|\Omega_p^\Delta|$. As an example, for $m = 100,000, \delta = 5\%$, a suitable approach to construct a sufficiently challenging target distribution is to divide the space into $|\Omega_p^\Delta| \leq 6$ sets. One of these ($\mathcal{S}_0 \in \Omega_p^\Delta$) should be very large, and is assigned 0 probability to meet the requirement that $|\Omega^+|/|\Omega| \approx 0$. The remaining sets ($\{\mathcal{S}_1, \dots\}$) can be of approximately equal size. We assign the same probability mass p_i to every element of set $\mathcal{S}_i$, i.e., $x \in \mathcal{S}_i \implies p_x = p_i$. We denote the highest and lowest pmf values by $p_{likely} \triangleq \max_i p_i$ and $p_{rare} \triangleq \min_i p_i$ and set the ratio p_{likely}/p_{rare} as a parameter of the synthetic distribution. This results in no single element having an unusually large probability.

Sequences. In practice, high dimensional categorical data of interest often appears in the form of sequences of categories (text, proteins). The generative models are tasked with learning the complex dependency structure within the sequences. To mimic such datasets, we map the categorical elements to sequences of length S, with each element of the sequence belonging to one of K categories. The dimensionality of this sequence's probability space is thus $K^S = |\Omega|$.

Following the procedure above, we assign the possible sequences to sets, with any sequences in $\mathcal{S}_0$ being assigned zero probability. Sequences are allocated to the other sets according to rules that specify the structure that the generative model should learn. The challenge can vary depending on the design. In our experiments, we focus on two examples. In the first, "PAIR", sequences are allocated to sets according to located pairwise occurrences of categories in the sequence. We define an arbitrary set of "invalid" subsequences of length $S = 2$ (for example $\{AA, AC, \dots\}$), and only assign positive probability to sequences that do not contain any of those subsequences. In the second, "PERM", we only allocate positive probability to sequences where no category appears twice, e.g., $ABCDEF$ or $FCDABE$. For both experiments, we define two sets with positive probability so we have $\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2$ with assigned probability values $0, p_{likely}, p_{rare}$. Further details concerning the exact construction of these examples are included in Appendix 4.

Learning from real data Although it is not the focus of our work, we propose an extension to apply a similar procedure to rank generative models when we

have access to experimental data and wish to construct a synthetic test distribution using that data. To do this, we train a surrogate generative model on the available samples. This surrogate offers direct evaluation of its learned pmf mass for any candidate element (sequence).

The testing process proceeds as follows. We train the surrogate model and denote by $\tilde{p}$ the pmf after training. We then generate a new dataset by sampling from $\tilde{p}$: $\mathcal{D}^S = \{\tilde{x}_i\}_{i=1}^m \quad \tilde{x}_i \sim \tilde{p}$ and define a new ground truth distribution as follows:

$$p_x^{new} = \begin{cases} \frac{\tilde{p}_x}{\sum_{x \in \mathcal{D}^S} \tilde{p}_x} & \text{if } x \in \mathcal{D}^S \\ 0 & \text{o.w.} \end{cases} \tag{12}$$

We can then follow our testing procedure using p^{new} as the synthetic ground truth distribution. We start by sampling a dataset from $\tilde{x} \sim p^{new}$ to train the generative models $q_1, q_2, \dots$ that we wish to evaluate. This process offers a trade-off between the synthetic and real scenario as our constructed ground truth p^{new} is based on what a model could learn from a real dataset. More details on this method and an example of using this procedure on a protein dataset and a Transformer generative model as the surrogate model are included in Appendix 1.

5 EXPERIMENTS

We start by validating our approach by conducting experiments on synthetic generative models for which the exact generative distribution is known. The generative models are built based on a ground truth distribution p parameterized with $|\mathcal{B}^1| = 5, \max_i p_i / \min_i p_i = 4$. We then introduce a controlled amount of total variation error ($d_{tv}(p, q_\epsilon) = \epsilon$) by modifying a fixed percent $b = 30\%$ of the support Ω^+ in two ways: q_ϵ^{flat}(FLAT) is obtained by modifying p_x elements across the whole space, and $q_\epsilon^{\uparrow\downarrow}$ (HIGH/LOW) is constructed by modifying either the low pmf values of p or the high pmf values of p with some probability. A complete description is included in Appendix 4.

We then present results on recently proposed categorical generative models by training them on sequence data. We select four SOTA generative categorical models: 1) **CNF** Lippe and Gavves (2021), a normalizing flow method with a learned mapping to the categorical space; 2) **CDM** Hoogeboom et al. (2021a), a discrete diffusion method based on Categorical distributions; 3) **argmaxAR** Hoogeboom et al. (2021b), a normalizing flow method with an argmax mapping; and 4) **GMCD** Regol and Coates (2023), a continuous diffusion method based on Gaussian Mixtures. We describe the architecture search and the training procedure in Appendix 5. We ensure that each baseline model has comparable training time and memory complexity[1].

5.1 Evaluation Baselines

In addition to our evaluation metrics, we report the NLL^m and an adaptation of coverage metrics to the categorical space. Although existing coverage metrics are not directly applicable in the categorical domain, we can use our knowledge of the structure of the problem to build a sensible distance function for the designed categorical sequences. Complete details regarding the distance metric are included in Appendix 4. We report the metrics IP_α, IR_β from Alaa et al. (2022) and Precision (Pr.) and Recall (R.) from Sajjadi et al. (2018) based on this adaptation from nominal to ordinal space. We note that the presented results do not reflect on the performance of the original metrics in their intended settings.

5.2 Results

For all results, we generate $m = 100,000$ samples that we split in 10 to perform a Wilcoxon ranking test to report statistical significance. We begin by validating our method with different synthetic generative models. All results have statistical significance compared with the next closest baselines at the 5% level. When the error introduced is the same across the whole space in the FLAT experiment, we see that all metrics are able to identify the correct ranking (Left of Table 1). For our metric, including both the main metric $\hat{\mathbf{T}}_7^*\%$ and the others, the synthetic generative model corresponding to the ground truth (p) has close to 0 error, correctly indicating that the model is very close to the true distribution. Using the bounds from Theorem 4.2, we can state with an 80% confidence that the baseline p is better than $q_{0.10}$ as their intervals do not overlap. A visualisation of the result can be seen in Appendix 8. The NLL evaluation also finds the correct ranking, but this is the only information that can be extracted from the metric values.

For the HIGH/LOW experiment where the error is not evenly distributed across the space, the evaluation is more challenging (Right of Table 1). For the NLL evaluation, the ranking is different from the total variation (highlighted in red in Table 1) with statistical significance. On the other hand, the proposed method still provides the correct ranking. We provide a visualization of the statistical intervals in Appendix 8. In this case, the overlap of the intervals does not allow us to make the same confidence statement.

For both experiments, the scale of the estimated total

[1]Code to reproduce our experiments is available `https://github.com/networkslab/eval_cat`

Table 1: FLAT (left) and HIGH/LOW (right) experiment, $|\Omega| = 10^{10}$, $m = 100,000$ samples, $|\Omega_p^{\Delta}| = 5$. In the HIGH/LOW experiment, The NLL^m provides a different ranking (red highlight).

	NLL^m	$\hat{T}_5\%$	$\hat{T}_6\%$	$\hat{\mathbf{T}}_{\mathbf{7}}^{*}\%$	$\hat{T}_8\%$	slope	OOD %	conc.%	NLL^m	$\hat{T}_5\%$	$\hat{T}_6\%$	$\hat{\mathbf{T}}_{\mathbf{7}}^{*}\%$	$\hat{T}_8\%$	slope	OOD %	conc.%
p	22.789	0.2	0.2	0.3	0.3	0.03	0	0.1	22.789	0.2	0.2	0.3	0.3	0.03	0	0.1
$q_{0.05}$	22.796	2.4	2.4	2.4	2.4	0.00	0.2	1.2	22.805	0.4	0.6	0.9	1.1	0.217	0.2	0.9
$q_{0.07}$	22.802	3.9	3.9	3.9	3.9	0.00	0.4	1.6	22.803	0.5	0.9	1.2	1.5	0.325	0.4	1.5
$q_{0.10}$	22.810	5.0	5.0	5.0	5.0	0.00	0.5	2.2	22.811	0.6	1.1	1.5	1.9	0.433	0.5	2.0

variation error in the $\mathcal{B}_i$ spaces is also of the same order as the true total variation error in the original space of 10 billion elements. By inspecting the slope of the increase of the estimated error across granularities, we observe that it is very close to zero for the FLAT experiment and higher for the HIGH/LOW experiment. This suggests that the error is spread relatively evenly across a large portion of the space for FLAT, but is more concentrated for HIGH/LOW. Indeed, this corresponds to the error generation mechanism.

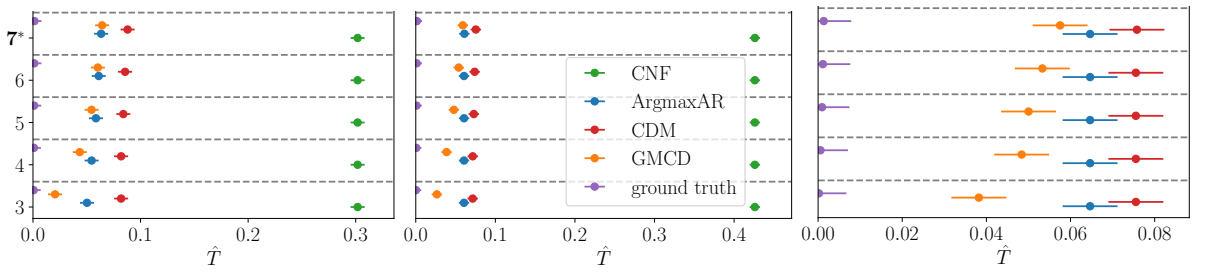
Real generative models Next we show how our procedure ranks existing generative models for the SEQUENCE experiments. Figure 5 displays the empirical pmfs of various models over a selected subset of the initial space Ω for a PERM experiment. Since we have no access to the ground truth ranking, we consider a smaller space ($|\Omega| = 6^6 \approx 50,000$) in order to be able to accurately estimate the true total variation. We report this as $\hat{d}_{tv}$, and use it as a proxy for the ground truth.

In this setting, we have to rely on the approximated NLL of the generative models. In Table 2, the NLL approximations do not provide a ranking with significance between argmaxAR and CDM, and GMCD does not provide log likelihood. In contrast, our metric provides a ranking that is consistent with the ranking provided by the total variation estimate over the whole space. Inspecting the intervals on the left of Figure 3, we can declare with 80% confidence that the GMCD and argmaxAR baselines have lower total variation error than the CDM and CNF models for the granularities considered $(3-7)$. Next, turning to the *OOD* metric, we can see that CDM and GMCD generate fewer out-of-samples than argmaxAR and CNF. Interestingly, CDM is the more robust to over-concentration for high probability elements. Lastly, we can see that the estimated slopes of the CDM and CNF are close to 0 indicating that the error is spread evenly across the space, which is confirmed by inspecting Figure 5. The GMCD and argmaxAR models have higher slopes, and Figure 5 confirms that the models have more outliers for this experiment. From 5, the CNF model struggles to assign different pmf values to different sets. This appears to be a characteristic of CNF as it performs even worse for the PERM experiment with a higher ratio $p_{likely}/p_{rare} = 7$ (center Figure 5), whereas the other baselines are largely unaffected. Lastly, although the GMCD baseline performs well overall, it has more outliers (see Figure 5), and this is revealed by the slope estimate, which is consistently high in all SEQUENCE experiments.

Turning to the coverage metrics, we see that these metrics do not clearly identify that the CNF is a much poorer model (Figure 4). This is to be expected as the metrics focus on coverage, and do not account for the probability mass assigned to specific elements. The CNF model does identify the positive support of the distribution just as well as the other models, but it fails to learn that some elements have higher probabilities. Similar trends can be observed for the experiments PAIR and PERM with a higher ratio p_{likely}/p_{rare} in Figure 5. The GMCD, CDM and ArgmaxAR models outperform CNF. ArgmaxAR and GMCD have higher slopes than CDM and CNF, suggesting that CDM and CNF have lower variance in their pmf value estimates around the ground truth pmf values. Lastly, we can see that all baseline performs better for the easier PAIR experiments.

6 LIMITATIONS

As we highlighted in Section 3.1, the main limitation of our work is that we cannot prove that the baseline that performs the best in our procedure has indeed the lowest total variation in the original space. Another limitation that directly stems from our setting is that it requires complete access to the ground truth p (synthetic), while many applications of interest for generative models only have limited sample access to p. The extension we proposed to apply our method to real data has some drawbacks — the task will inevitably be heavily skewed towards the surrogate generative model used as a basis for the new ground truth distribution. This means that our evaluation procedure cannot be used to evaluate the surrogate model itself, or models with similar architecture. Moreover, this will also increase the time complexity of our test as we will have to iterate over the new space $|\Omega^{new}| \approx m$ to find $\Omega_{p^{new}}^{\Delta}$. That being said, we still believe that it provides



PERM, $p_{likely}/p_{rare} = 3$ PERM, $p_{likely}/p_{rare} = 7$ PAIR, $p_{likely}/p_{rare} = 7$

Figure 3: Probabilistic intervals from Theorem 4.1. with $\delta = 10\%$, $m = 100,000$ samples for 3 SEQUENCE experiments, at varying granularities. If the intervals of two baselines do not overlap at a given granularity, we can state with 80% confidence that the baseline with the lowest total variation estimator $\hat{T}$ has lower total variation than the other baseline at that granularity.

Table 2: PERM exp., $p_{likely}/p_{rare} = 3$. A star (*) indicates statistical significance compared with the next closest baseline at the 5% level.

	$\hat{d}_{tv}$	NLL	$\hat{T}_3$	$\hat{T}_4$	$\hat{T}_5$	$\hat{T}_6$	$\hat{\mathbf{T}}_7^*$	$\hat{T}_8$	slope	OOD %	conc.%
GMCD	**0.149**$*$	-	3.89*	4.36*	4.70*	4.98*	**5.20***	5.32*	0.284	**0.8**	0.8
argmaxAR	0.171*	**1.1104 ±0.1200**	6.66*	7.28*	7.90*	8.40*	8.78*	8.90*	0.463	6.3	1.2
CDM	0.183*	1.1111 ±0.1446	10.14*	10.14*	10.14*	10.14*	10.14*	10.16*	0.003	1.0	**0.7**
CNF	0.281*	1.1658* ±0.1167	27.70*	27.70*	27.70*	27.70*	27.70*	27.70*	0.000	5.1	9.3

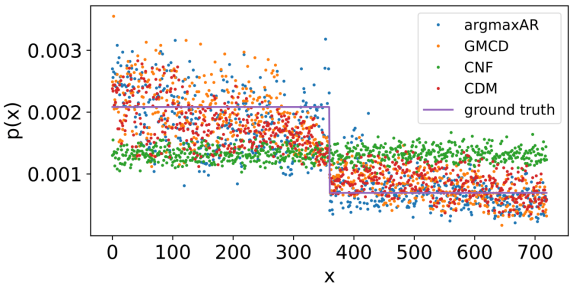


Figure 4: Empirical pmf q of generative models on Ω^+ with sorted ground truth.

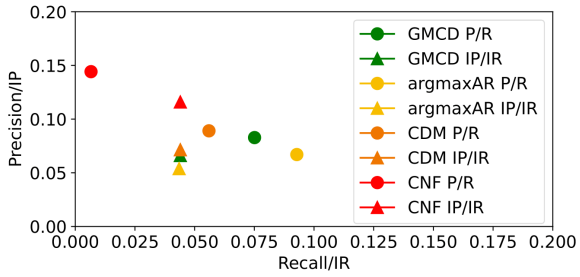


Figure 5: Precision/Recall, and IP_α/ IR_β. Our adaptation of the ordinal coverage metrics to a nominal categorical space is insufficient to render these metrics functional in our problem setting.

a good trade-off between our purely synthetic proposal and a sampled-based approach. We stress that while the synthetic setting is a constraint in our approach, when it comes to addressing the evaluation problem to the extent that we have, every metric would encounter comparable or more significant limitations.

7 CONCLUSION

We have introduced a principled and robust method for assessing the performance of categorical generative models in a synthetic setting, where the target ground truth is constructed to resemble a realistic task. Our metric is accurate, interpretable, scalable to high dimensional settings, and robust to different mode of failure of the generative model. This is supported by experiments using both controlled generative models with known behaviour, as well as state-of-the-art categorical generative models. A promising avenue for future work is to address the main limitation of our work and propose an improved version that can be applied to real datasets without requiring explicit access to the ground truth p.

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., and Nahavandi, S. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297.

Alaa, A., Van Breugel, B., Saveliev, E. S., and van der Schaar, M. (2022). How faithful is your synthetic data? Sample-level metrics for evaluating and auditing generative models. In *Proc. Int. Conf. Machine Learning (ICML)*.

Batu, T., Fischer, E., Fortnow, L., Kumar, R., Rubinfeld, R., and White, P. (2001). Testing random variables for independence and identity. In *Proc. IEEE Symp. on Foundations of Computer Sci.*

Borji, A. (2019). Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65.

Caccia, M., Caccia, L., Fedus, W., Larochelle, H., Pineau, J., and Charlin, L. (2020). Language GANs falling short. In *Proc. Int. Conf. Learning Representations (ICLR)*.

Canonne, C. L. (2020a). A short note on learning discrete distributions. arXiv preprint: arXiv 2002.11457.

Canonne, C. L. (2020b). A survey on distribution testing: Your data is big. but is it blue? *Electron. Colloquium Comput. Complex.*, TR15(9):1–100.

Canonne, C. L. (2022). Topics and techniques in distribution testing: A biased but representative sample. *Foundations and Trends in Communications and Information Theory*, 19(6):1032–1198.

Celikyilmaz, A., Clark, E., and Gao, J. (2020). Evaluation of text generation: A survey. arXiv preprint: arXiv 2006.14799.

Diakonikolas, I., Gouleakis, T., Kane, D. M., Peebles, J., and Price, E. (2021). Optimal testing of discrete distributions with high probability. In *Proc. ACM SIGACT Symp. on Theory of Computing*, page 542–555.

Djolonga, J., Lucic, M., Cuturi, M., Bachem, O., Bousquet, O., and Gelly, S. (2020). Precision-recall curves using information divergence frontiers. In *Proc. Int. Conf. Artificial Intelligence and Statistics (AISTAT)*.

El-Gebali, S., Mistry, J., Bateman, A., Eddy, S. R., Luciani, A., Potter, S. C., Qureshi, M., Richardson, L. J., Salazar, G. A., Smart, A., Sonnhammer, E. L. L., Hirsh, L., Paladin, L., Piovesan, D., Tosatto, S. C. E., and Finn, R. D. (2019). The pfam protein families database in 2019. *Nucleic Acids Res.*, 47(D1):D427–D432.

Garbacea, C., Carton, S., Yan, S., and Mei, Q. (2019). Judge the judges: A large-scale evaluation study of neural language models for online review generation. In *Proc. Conf. on Empirical Methods in Natural Language Process. and Int. Joint Conf. on Natural Language Process (EMNLP-IJCNLP)*.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Proc. Adv. Neural Info. Process. Syst. (NeurIPS)*.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. (2021a). Argmax flows and multinomial diffusion: Learning categorical distributions. In *Proc. Adv. Neural Info. Process. Syst. (NeurIPS)*.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. (2021b). Argmax flows: Learning categorical distributions with normalizing flows. In *Proc. Symp. on Adv. in Appr. Bayesian Inference*.

Hua, W., Dai, Z., Liu, H., and Le, Q. (2022). Transformer quality in linear time. In *Proc. Int. Conf. Machine Learning (ICML)*.

Jiralerspong, M., Bose, J., Gemp, I., Qin, C., Bachrach, Y., and Gidel, G. (2023). Feature likelihood score: Evaluating the generalization of generative models using samples. In *Proc. Adv. Neural Info. Process. Syst. (NeurIPS)*.

Khayatkhoei, M. and AbdAlmageed, W. (2023). Emergent asymmetry of precision and recall for measuring fidelity and diversity of generative models in high dimensions. In *Proc. Int. Conf. Machine Learning (ICML)*.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. (2019). Improved precision and recall metric for assessing generative models. In *Proc. Adv. Neural Info. Process. Syst. (NeurIPS)*.

Lippe, P. and Gavves, E. (2021). Categorical normalizing flows via continuous transformations. In *Proc. Int. Conf. Learning Representations, (ICLR)*.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. (2020). On the variance of the adaptive learning rate and beyond. In *Proc. Int. Conf. Learning Representations (ICLR)*.

Liu, L., Pillutla, K., Welleck, S., Oh, S., Choi, Y., and Harchaoui, Z. (2021). Divergence frontiers for generative models: Sample complexity, quantization effects, and frontier integrals. In *Proc. Adv. Neural Info. Process. Syst. (NeurIPS)*.

Naeem, M. F., Oh, S. J., Uh, Y., Choi, Y., and Yoo, J. (2020). Reliable fidelity and diversity metrics for generative models. In *Proc. Int. Conf. Machine Learning (ICML)*.

Nagarajan, V., Andreassen, A., and Neyshabur, B. (2021). Understanding the failure modes of out-of-distribution generalization. In *Proc. Int. Conf. Learning Representations (ICLR)*.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. (2019). Do deep generative models know what they don't know? In *Proc. Int. Conf. Learning Representations (ICLR)*.

Novikova, J., Dušek, O., Cercas Curry, A., and Rieser, V. (2017). Why we need new evaluation metrics for NLG. In *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*.

OpenAI (2023). Gpt-4 technical report.

Regol, F. and Coates, M. (2023). Diffusing gaussian mixtures for generating categorical data. In *Proc. Ass. for the Adv. of Artificial Intelligence (AAAI)*.

Regol, F., Kroon, A., and Coates, M. (2023). Evaluation of categorical generative models - bridging the gap between real and synthetic data. In *Proc. EEEI Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*.

Sajjadi, M. S. M., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. (2018). Assessing generative models via precision and recall. In *Proc. Adv. Neural Info. Process. Syst. (NeurIPS)*.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., and Chen, X. (2016). Improved techniques for training GANs. In *Proc. Adv. Neural Info. Process. Syst. (NeurIPS)*.

Shibuya, T., Takida, Y., and Mitsufuji, Y. (2023). BigVSAN: Enhancing GAN-based neural vocoders with slicing adversarial network. arXiv preprint: arXiv 2309.02836.

Shumailov, I., Shumaylov, Z., Zhao, Y., Gal, Y., Papernot, N., and Anderson, R. (2023). The curse of recursion: Training on generated data makes models forget. arXiv preprint: arXiv 2305.17493.

Theis, L., van den Oord, A., and Bethge, M. (2016). A note on the evaluation of generative models. In *Proc. Int. Conf. Learning Representations (ICLR)*.

Thompson, R., Knyazev, B., Ghalebi, E., Kim, J., and Taylor, G. W. (2022). On evaluation metrics for graph generative models. In *Proc. Int. Conf. Learning Representations (ICLR)*.

Valiant, G. and Valiant, P. (2017). An automatic inequality prover and instance optimal identity testing. *SIAM Journ. on Computing*, 46(1):429–455.

van den Oord, A. and Schrauwen, B. (2014). Factoring variations in natural images with deep gaussian mixture models. In *Proc. Adv. Neural Info. Process. Syst. (NeurIPS)*.

Veselovsky, V., Ribeiro, M. H., Arora, A., Josifoski, M., Anderson, A., and West, R. (2023). Generating faithful synthetic data with large language models: A case study in computational social science. arXiv preprint: arXiv 2305.15041.

Wang, J., HU, X., Hou, W., Chen, H., Zheng, R., Wang, Y., Yang, L., Ye, W., Huang, H., Geng, X., Jiao, B., Zhang, Y., and Xie, X. (2023). On the robustness of chatGPT: An adversarial and out-of-distribution perspective. In *Proc. Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models (ICLR)*.

Wu, Y., Burda, Y., Salakhutdinov, R., and Grosse, R. B. (2017). On the quantitative analysis of decoder-based generative models. In *Proc. Int. Conf. Learning Representations (ICLR)*.

Yang, J., Wang, P., Zou, D., Zhou, Z., Ding, K., PENG, W., Wang, H., Chen, G., Li, B., Sun, Y., Du, X., Zhou, K., Zhang, W., Hendrycks, D., Li, Y., and Liu, Z. (2022). OpenOOD: Benchmarking generalized out-of-distribution detection. In *Proc. Adv. Neural Info. Process. Syst. (NeurIPS)*.

Yu, L., Cheng, Y., Sohn, K., Lezama, J., Zhang, H., Chang, H., Hauptmann, A. G., Yang, M.-H., Hao, Y., Essa, I., and Jiang, L. (2023). MAGVIT: Masked generative video transformer. In *Proc. IEEE/CVF Conf. Comp. Vision and Patt. Recognition (CVPR)*.

Zhou, S., Gordon, M. L., Krishna, R., Narcomey, A., Fei-Fei, L., and Bernstein, M. S. (2019). HYPE: A benchmark for human eye perceptual evaluation of generative models. In *Proc. Adv. Neural Info. Process. Syst. (NeurIPS)*.

CHECKLIST

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:

(a) Statements of the full set of assumptions of all theoretical results. [Yes]
(b) Complete proofs of all theoretical results. [Yes]
(c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

SUPPLEMENTARY MATERIAL

1 EXTENSION TO REAL DATA

We propose a method to extend our approach to real data by using the probability mass function (pmf) learned by a surrogate model on a real dataset as a proxy for the distribution of the real data. Admittedly, the proxy distribution can only be as good as the surrogate model itself, but it is not necessary for the surrogate model to be extremely accurate for this test to be useful.

The benefit of this approach is that the nature of the sequence and the complexity can more closely resemble a practical scenario than an artificial task. As an example, we present how this method can be used on protein datasets from the Pfam protein family PF00014 using the Transformer from Hua et al. (2022) as a surrogate model. Protein datasets contain sequences of amino acids (there are 21 amino acids, and hence, $K = 21$). This specific dataset contains sequences of length $S = 53$ and has a training/testing/validation set of $10,000$ samples each.

We start by training an Autoregressive Transformer for 1000 epochs. Table 4 presents the hyperparameters. As we stated in Section 4, "Learning from real data," in the main text, we denote by $\tilde{p}$ the pmf of the transformer after training. We then generate a new dataset by sampling 100,000 samples from $\tilde{p}$: $\mathcal{D}^S = \{\tilde{x}_i\}_{i=1}^{100,000} \quad \tilde{x}_i \sim \tilde{p}$ and define a new ground truth distribution as follows:

$$p_x^{new} = \begin{cases} \frac{\tilde{p}_x}{\sum_{x \in \mathcal{D}^S} \tilde{p}_x} & \text{if } x \in \mathcal{D}^S \\ 0 & \text{o.w.} \end{cases} \tag{13}$$

Since we have no control over the ground truth pmf p^{new}, the initial set $\Omega_{p^{new}}^{\Delta}$ can become severely unbalanced if low and high pmf values are concentrated in a few elements. In practice, we observe that this is the case in this experiment. Instead of following Eqn. 2 from the main text directly to define $\mathcal{B}^1 = \Omega_p^{\Delta}$ as:

$$\Omega_p^{\Delta} = \underset{\mathcal{X} \in \rho_p^{\Delta}(\Omega)}{\arg\min} |\mathcal{X}|, \tag{14}$$

$$\rho_p^{\Delta}(\Omega) = \{\{\mathcal{S}_1^{\Delta}, \dots\} \, s.t. \, |p_{x_j} - p_{x_k}| \leq \Delta \quad \forall x_j, x_k \in \mathcal{S}_i^{\Delta}\}, \tag{15}$$

Table 3: Proxy experiment, $|\Omega| = 21^{53}$, $m = 100,000$ samples, $|\Omega_p^{\Delta}| = 4$.

	$\hat{T}_4$	$\hat{T}_5$	$\hat{T}_6$	$\hat{T}_7$
CDM	0.279	0.421	0.537	0.631
GMCD	0.244	0.402	0.529	0.628
ground truth	0.000	0.001	0.001	0.002

we instead order the elements with positive mass by their pmf mass values: $x_1, x_2, \dots; 0 < p(x_i) < p(x_{i+1})$,

and split this ordered sequence in s sets that each have the same probability mass:

$$\Omega_p^s = \{\mathcal{S}_1, \dots, \mathcal{S}_s; p(\mathcal{S}_i) = 1/s\} \cup \{\mathcal{S}_0; p(\mathcal{S}_0) = 0\}. \tag{16}$$

We set the value of s to 3 in our experiment. We show the results for that experiment in Table 3. As the granularity increases, we can see that the total error variation increases drastically, suggesting that the true error in the original space is probably very large. This is to be expected as it is a very hard task and the target p^{new} not only has small support, but is very concentrated within that support.

Table 4: Transformer hyperparameters. If not listed in the table, the hyperparameter value takes the default value from Hua et al. (2022).

Hyperparameters	**PF00014**
depht	6
size	128
causal	True
batch size	32
training iterations	1k
learning rate	1e-4

2 TOTAL VARIATION ERROR OF PARTITIONED SPACES

In the methodology, we state that the total variation on a partitioned space $d_{tv}(p^{\mathcal{B}}, q^{\mathcal{B}})$ is always greater or equal than the total variation in the original space $d_{tv}(p, q)$. We provide a simple proof for that statement.

Recall that $\rho(\Omega)$ denotes the set containing every partition of the initial probability space Ω. The induced pmf $p^{\mathcal{B}}_{\mathcal{A}}$ $\mathcal{A} \in \mathcal{B}$ is obtained from a distribution p defined over Ω and a partitioning of Ω $\mathcal{B} \in \rho(\Omega)$ by summing the probability mass of each elements contained in the set $\mathcal{A}$:$p^{\mathcal{B}}_{\mathcal{A}} = \sum_{x \in \mathcal{A}} p_x$.

Lemma 1 : $\mathcal{B} \in \rho(\Omega) \implies d_{TV}(p^{\mathcal{B}}, q^{\mathcal{B}}) \leq d_{TV}(p, q)$.

Proof.

$$d_{TV}(p^{\mathcal{B}}, q^{\mathcal{B}}) = \frac{1}{2} \sum_{\mathcal{A}_i \in \mathcal{B}} |p^{\mathcal{B}}_{\mathcal{A}_i} - q^{\mathcal{B}}_{\mathcal{A}_i}| \tag{17}$$

$$= \frac{1}{2} \sum_{\mathcal{A}_i \in \mathcal{B}} |\sum_{x \in \mathcal{A}_i} p_x - \sum_{x \in \mathcal{A}_i} q_x| \text{ by def. of } \mathcal{B} \tag{18}$$

$$\leq \frac{1}{2} \sum_{\mathcal{A}_i \in \mathcal{B}} \sum_{x \in \mathcal{A}_i} |p_x - q_x| \text{ by the triangle ineq.} \tag{19}$$

$$= \frac{1}{2} \sum_{\mathcal{A}_i \in \mathcal{B}} \sum_{x \in \mathcal{A}_i} |p_x - q_x| \tag{20}$$

$$= \frac{1}{2} \sum_{x \in \Omega} |p_x - q_x| \tag{21}$$

$$d_{TV}(p^{\mathcal{B}}, q^{\mathcal{B}}) \leq d_{TV}(p, q) \tag{22}$$

□

3 NEAR-Δ PARTITIONING

Algorithm 2 near-Δ set Ω_p^Δ

Input: Distribution p with associated sample space Ω, tolerance parameter Δ.
Initialize $\Omega_p^\Delta = \{\}$
Initialize reminder $\mathcal{R} = \Omega$
while $\mathcal{R} \neq \emptyset$ **do**
  Set $p_{max} = \max_{x \in \mathcal{R}} p_x$
  $\mathcal{S}^\Delta = \{x \in \mathcal{R}; p_x > p_{max} - \Delta\}$
  $\mathcal{R} = \mathcal{R} \setminus \mathcal{S}$
  $\Omega_p^\Delta = \Omega_p^\Delta \cup \mathcal{S}^\Delta$
end while
Return Ω_p^Δ

Given a distribution p and a tolerance parameter $\Delta \in [0, 1)$, our procedure to obtain a near-Δ set Ω_p^Δ is described in Algorithm 2. Recall the definition of Ω_p^Δ given in Eqn. 2 in the main text:

$$\Omega_p^\Delta = \underset{\mathcal{X} \in \rho_p^\Delta(\Omega)}{\arg\min} |\mathcal{X}|, \tag{23}$$

$$\rho_p^\Delta(\Omega) = \{\{\mathcal{S}_1^\Delta, \dots\} \, s.t. \, |p_{x_j} - p_{x_k}| \leq \Delta \quad \forall x_j, x_k \in \mathcal{S}_i^\Delta\}. \tag{24}$$

We show how to obtain this partitioning in Algorithm 2.

Lemma 2: Given that Ω_p^Δ is obtained from Algorithm 2, there is no other $\Omega_p'^\Delta$ s.t. $|\Omega_p'^\Delta| < |\Omega_p^\Delta|$.

Proof. Assuming Ω_p^Δ is obtained from Algorithm 2 and has cardinality $s = |\Omega_p^\Delta|$, we can define an element with maximum probability mass in each set of Ω_p^Δ :

$\{x^i_{max} = \arg\max_{x \in \mathcal{S}^\Delta_i} p_x\}^s_{i=1}$. So each set in $\mathcal{S}^\Delta_i \in \Omega^\Delta_p$ have it's corresponding maximum pmf element x^i_{max}. In short, $p(x) \leq p(x^i_{max}) \forall x \in \mathcal{S}^\Delta_i$.

By lines 5-6 of Algorithm 2, we have that the pmf discrepancy between two different x^i_{max} and x^j_{max} must be higher than Δ, because otherwise they would be placed in the same set. Thus $|p_{x^i_{max}} - p_{x^j_{max}}| > \Delta \, \forall i \neq j \in \{1, s\}$. Hence, any two x^i_{max}, x^j_{max} cannot be placed in the same set $\mathcal{S}^\Delta$ to form a Ω'^Δ_p by the definition in Eqn. (2) (in the main text). Since we have s different x^i_{max}, then this implies that we need at least s sets to form any other solution Ω'^Δ_p . Hence, $|\Omega'^\Delta_p| \geq |\Omega^\Delta_p|$ for any other Ω'^Δ_p, which concludes the proof. □

4 GROUND TRUTH CONSTRUCTION

In this section, we give the complete description of both the construction of the ground truth p for the synthetic and sequence experiments, and for the construction of generative models $q^{flat}, q^{\uparrow\downarrow}$ for the synthetic experiments.

We start by restating the requirements for p that were laid out in Section 4 (main text):

$$p(x) = \begin{cases} 0, & \text{if } x \in \mathcal{S}_0\,, \\ p_i, & \text{if } x \in \mathcal{S}_i \text{ for } i > 0\,. \end{cases} \tag{25}$$

$\Omega^+ = \cup_{i=1} \mathcal{S}_i$ is the support of the pmf and we have that $|\Omega^+|/|\Omega| \approx 0$, hence $\mathcal{S}_0$ is very large. For simplicity, we set the support sets to be of equal size $|\mathcal{S}_i| = |\mathcal{S}_j|, \forall i, j > 0$ and assign probability mass values that are linearly increasing between p_{min} and p_{max}. We set Δ to approach zero for Ω^Δ_p, and hence we have that the solution $\Omega^{\Delta \to 0}_p$ coincides exactly with the specified sets $\{\mathcal{S}_0, \mathcal{S}_1, \dots\}$.

4.1 Synthetic experiment

Parameters of the ground truth p. In the synthetic experiment, we set the ratio p_{likely}/p_{rare} to 5 and split the space in 5 :$\{\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$. We assign 97% of the elements in Ω to $\mathcal{S}_0$ and split the remaining elements evenly between $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\}$.

Parameters of the synthetic generative model q^{flat}. : The distribution q^{flat}_ϵ is constructed by adding $\epsilon/b|\Omega^+|$ probability mass to the $b|\Omega^+|/2$ highest probability elements, and removing the same amount from the $b|\Omega^+|/2$ lowest probability elements. We set b to 30%.

Parameters of the synthetic generative model $q^{\uparrow\downarrow}$. The distribution $q^{\uparrow\downarrow}_\epsilon$ is constructed by adding $2\epsilon/b|\Omega^+|$ probability mass to $b/4|\Omega^+|$ elements, and removing the same amount from another $b/4|\Omega^+|$ elements. In total, this perturbs $b/2|\Omega^+|$ elements. With probability 1/2, these elements are randomly drawn from the $b|\Omega^+|/2$ highest probability elements, and with probability 1/2, these elements are randomly drawn from the $b|\Omega^+|/2$ lowest probability elements. We set b to 30%.

4.2 SEQUENCES

To generate the sequence experiment to train real generative models, we map elements in Ω to sequences. We build the space of sequences of categories $\Omega = \{C_1, C_2, \dots, C_K\}^K$, where $\{C_1, C_2, \dots, C_K\}$ is a set of categories. For simplicity, we set the number of categories to be the same as the length of the sequence $S = K$. A sequence is denoted by $\mathbf{x} \in \Omega$ and the i-th element of that sequence is denoted by $\mathbf{x}_i \in \{C_1, C_2, \dots, C_K\}$. As we describe in the main text, we construct the ground truth pmf p by partitioning the space and assigning the same probability to each element belonging to the same set. For simplicity, we split the space in 3 partitions: $\mathcal{S}_2, \mathcal{S}_1, \mathcal{S}_0$ and assign $p(x) = p_{likely} \, \forall x \in \mathcal{S}_2$, $p(x) = p_{rare} \, \forall x \in \mathcal{S}_1$ and $p(x) = 0 \; \forall x \in \mathcal{S}_0$. We set the ratio p_{likely}/p_{rare} as a parameter and obtain the pmf values by solving $p_{likely}|\mathcal{S}_{likely}| + p_{rare}|\mathcal{S}_{rare}| = 1$. The pmf is given by:

$$p(\mathbf{x}) = \begin{cases} p_{likely}, & \text{if } \mathbf{x} \in \mathcal{S}_2, \\ p_{rare}, & \text{if } \mathbf{x} \in \mathcal{S}_1, \\ 0, & \text{otherwise}\,. \end{cases} \tag{26}$$

PERM In the PERM experiment, we only assign probability mass to permutations of $\{C_1, C_2, \dots, C_K\}$. Hence Ω^+ is the set of all permutations. To determine which permutation goes into which set ($\mathcal{S}_2$ or $\mathcal{S}_1$), we induce an artificial ordering in the categories: $C_i < C_j$ if $i < j$, and place all permutations that have $\mathbf{x}_1 < \mathbf{x}_K$ in $\mathcal{S}_2$.

$$p(\mathbf{x}) = \begin{cases} p_{likely}, & \text{if } \mathbf{x} \in \Omega^+ \wedge \mathbf{x}_1 < \mathbf{x}_K, \\ p_{rare}, & \text{if } \mathbf{x} \in \Omega^+ \wedge \mathbf{x}_1 > \mathbf{x}_K, \\ 0, & \text{otherwise}\,. \end{cases} \tag{27}$$

Hence we have that $|\mathcal{S}_2| = |\mathcal{S}_1| = K!/2$, $|\Omega^+| = K!$ and $|\Omega| = K^K$. As K grows, the ratio $|\Omega^+|/|\Omega|$ approaches zero as required.

PAIR For this simpler task, we base the construction on pairwise correlation only. We define an arbitrary set of "invalid" subsequences of length $S = 2$ and only assign positive probability to sequences that do not contain those sequences. To do so, we again induce an artificial ordering in the categories: $C_i < C_j$ if $i < j$, and given $\mathbf{x}_i = C_i$, the choices for $\mathbf{x}_{i+1}$ are in

the set $\{C_{i\%K}, \dots, C_{i+K/2\%K}\}$ where % denotes the modulo operator. For example, assuming the ordering $A < B < C < D$, the element y in a sequence $xAyx$ can only take the category $y = A$ or $y = B$. To determine which valid sequence goes into which set ($\mathcal{S}_2$ or $\mathcal{S}_1$), we further assign an integer value that matches the arbitrary ordering ($A = 1 < B = 2 < C = 3 < D = 4$) and place all valid sequences that have $\mathbf{x}_1 + \mathbf{x}_K \% 2 = 0$ in $\mathcal{S}_2$.

$$p(\mathbf{x}) = \begin{cases} p_{likely}, & \text{if } \mathbf{x} \in A_{likely} \\ p_{rare}, & \text{if } \mathbf{x} \in A_{rare} \\ 0, & \text{otherwise}. \end{cases} \tag{28}$$

$$\text{where } A_{likely} = \{\mathbf{x}; \mathbf{x} \in A_+ \wedge \mathbf{x}_1 + \mathbf{x}_K \% 2 = 0\}, \tag{29}$$

$$A_{rare} = \{\mathbf{x}; \mathbf{x} \in A_+ \wedge \mathbf{x}_1 + \mathbf{x}_K \% 2 = 1\}. \tag{30}$$

4.3 Coverage metrics details

We build a distance that clusters the sequences belonging to the same group (A_{likely}, A_{rare} or $\Omega \setminus A_+$) together. To avoid collapsing all sequences to the same point, we add a small contribution based on the Hamiltonian distance denoted by $H(\cdot, \cdot)$.

Hence our distance between two sequences is given as follows;

$$d(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 0.01H(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \mathbf{x}_i, \mathbf{x}_j \in A_{lik.}, \\ 0.01H(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \mathbf{x}_i, \mathbf{x}_j \in A_{rare} \\ 0.01H(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \mathbf{x}_i, \mathbf{x}_j \notin A_+ \\ 1 + 0.01H(\mathbf{x}_i, \mathbf{x}_j) & \text{if } \mathbf{x}_{i/j} \in A_{lik.}, A_{rare} \\ 4 + 0.01H(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \mathbf{x}_{i/j} \in A_+, \notin A_+ . \end{cases} \tag{31}$$

We provide the numeral values from the right Figure 4 in the main text in Table 5. The various coverage metrics do not correlate at all with the rank given by ground truth d_{tv} metric; our adaption of the coverage metrics to be applicable to a nominal setting is insufficient to obtain functional metrics for our setting.

5 PROOFS

Theorem 5.1. *Given a discrete distribution p with associated sample space $\mathcal{B}$, m samples from a distribution q with the same sample space $\mathcal{B}$, and an error tolerance $\epsilon_{test} \in (0, 1]$, provided that:*

$$\epsilon_{test} \geq \max\left(\sqrt{\frac{|\mathcal{B}|}{m}}, \sqrt{\frac{2\ln(2/\delta)}{m}}\right). \tag{32}$$

we can be at least $1-\delta$ confident that the true total variation $d_{TV}(p, q)$ is within the interval $[\hat{T}^m - \epsilon_{test}, \hat{T}^m + \epsilon_{test}]$.

Proof. Let $\tilde{q}$ denote the empirical distribution obtained from the samples of q. By the triangle inequality:

$$d_{TV}(p, \tilde{q}) - d_{TV}(\tilde{q}, q) \tag{33}$$

$$\leq d_{TV}(p, q) \leq d_{TV}(p, \tilde{q}) + d_{TV}(\tilde{q}, q) \tag{34}$$

Theorem 1 of Canonne (2020a) shows that $d_{TV}(q, \tilde{q}) \leq \epsilon_{\text{test}}$ with at least probability $1 - \delta$ if (32) holds.

Hence, with at least probability $1 - \delta$;

$$d_{TV}(p, \tilde{q}) - \epsilon_{\text{test}} \leq d_{TV}(p, q) \leq d_{TV}(p, \tilde{q}) + \epsilon_{\text{test}}\,, \tag{35}$$

$$\hat{T}^m - \epsilon_{\text{test}} \leq d_{TV}(p, q) \leq \hat{T}^m + \epsilon_{\text{test}}\,. \tag{36}$$

$$\text{which implies } \hat{T}^m \leq d_{TV}(p, q) + \epsilon_{\text{test}} \tag{37}$$

$$\text{and } \hat{T}^m \geq d_{TV}(p, q) - \epsilon_{\text{test}}\,. \tag{38}$$

Joining those two inequalities together gives us the interval $\hat{T}^m \in [d_{tv}(p, q) - \epsilon_{\text{test}}, d_{tv}(p, q) + \epsilon_{\text{test}}]$, which concludes the proof. □

Theorem 5.2. *Given a discrete distribution p with associated sample space $\mathcal{B}$, and m samples from the distributions q and q' with the same sample space $\mathcal{B}$, denote by $\hat{T}^m_q$ and $\hat{T}^m_{q'}$ the empirical total variation estimators of q and q', respectively. For an error tolerance $\epsilon_{test} \in (0, 1]$ s.t. (32) holds for a selected constant $\delta \in (0, 1)$, the random quantity $\hat{T}^m_q - \hat{T}^m_{q'}$ will fall within the following interval:*

$$\hat{T}^m_q - \hat{T}^m_{q'} \in [d_{tv}(p, q) - d_{tv}(p, q') \pm 2\epsilon_{\text{test}}] \tag{39}$$

with at least $(1 - \delta)^2$ probability.

Proof. Assuming that $\tilde{B}^m_q \in [d_{tv}(p, q) - \epsilon_{\text{test}}, d_{tv}(p, q) + \epsilon_{\text{test}}]$ and that $\hat{T}^m_{q'} \in [d_{tv}(p, q') - \epsilon_{\text{test}}, d_{tv}(p, q') + \epsilon_{\text{test}}]$ both holds;

$$\hat{T}^m_q - B^m_{q'} \leq d_{tv}(p, q) - d_{tv}(p, q') + 2\epsilon_{\text{test}}\,, \tag{40}$$

$$\text{and } \hat{T}^m_q - B^m_{q'} \geq d_{tv}(p, q) - d_{tv}(p, q') - 2\epsilon_{\text{test}}. \tag{41}$$

Since the two events (event $\tilde{B}^m_q \in [d_{tv}(p, q) \pm \epsilon_{\text{test}}]$ and event $\hat{T}^m_{q'} \in [d_{tv}(p, q') \pm \epsilon_{\text{test}}]$) are independent, the probability that both events occur is given by $P(\tilde{B}^m_q \in [d_{tv}(p, q) \pm \epsilon_{\text{test}}]) P(\hat{T}^m_{q'} \in [d_{tv}(p, q') \pm \epsilon_{\text{test}}]) \geq (1 - \delta)^2$ (by Theorem 3.1).

□

6 TRAINING PROCEDURE

All generative models are trained with the RAdam optimizer Liu et al. (2020) with a learning rate decay of 0.999975, and parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We use early stopping with a patience of 50 epochs

Table 5: Ranking performance of d_{tv} compared to Precision/Recall, IP_α/ IR_β and Authenticity for the SEQUENCE experiment.

	d_{TV} rank	Pr.	R.	IP_α	IR_β	Authen.
GMCD	1	0.083 (3)	0.075 (3)	0.066 (3)	0.044 (1)	0.0003 (4)
argmaxAR	2	0.067 (4)	0.093 (4)	0.054 (4)	0.043 (4)	0.0008 (1)
CDM	3	0.089 (2)	0.056 (2)	0.072 (2)	0.044 (1)	0.0004 (2)
CNF	4	0.144 (2)	0.007 (1)	0.116 (2)	0.044 (1)	0.0004 (2)

and select the model with the best evaluation over a validation set (10%). Tables 6, 7, 8, and 9 report the selected architecture parameters for GMCD, CMD, CNF and argmaxAR and the grid search values. As stated in the main text, we ensured that the model architecture and training time were not significantly different, unless achieving reasonable performance was unattainable.

Table 6: GMCD hyperparameters.

Hyperparameters	$K = 6$
dim of $\mathbf{Z}$ (d)	$\{3, \ldots, \underline{6}\}$
hidden size	{ 16, 32, $\underline{64}$}
num. heads	8
depth	2
num. blocks	{ 1, $\underline{2}$}
local size	64
local heads	4
dropout	0.2
T	{$\underline{10}$, ..., 50}
batch size	1024
training iterations	1k
learning rate	{7.5e-3, $\underline{7.5\text{e-}4}$, 7.5e-5}

7 DETAILED DISCUSSION OF THE PRINCIPLE LIMITATION

In our proposal, we highlighted that the principle limitation of our approach is that in general, $d_{tv}(p^{\mathcal{B}}, q_1^{\mathcal{B}}) \leq d_{tv}(p^{\mathcal{B}}, q_2^{\mathcal{B}})$ does not imply that $d_{tv}(p, q_1) \leq d_{tv}(p, q_2)$. This is actually to be expected, because otherwise we would be violating the optimal bound found in Diakonikolas et al. (2021). In this section, we study this undesirable case of $d_{tv}(p^{\mathcal{B}}, q_1^{\mathcal{B}}) \leq d_{tv}(p^{\mathcal{B}}, q_2^{\mathcal{B}})$ and $d_{tv}(p, q_1) \leq d_{tv}(p, q_2)$. We identify the requirements on $p, q_1, q_2, \mathcal{B}$ for this to occur, and this leads us to highlight the implicit bias of our method by exposing which types of generative models are favorably treated by our approach.

We start by decomposing the total variation error between p, q in the original space. The space can be split into two sets; one containing the elements where the generative model q overestimates p: $\Omega^+ \triangleq \{x \in \Omega \, s.t \, q_x \geq p_x\}$ (overestimation error e_x^+) and another containing the underestimated elements: $\Omega^+ \triangleq \{x \in \Omega \, s.t \, q_x < p_x\}$ (underestimation error e_x^-). The total variation can be decomposed into contributions from those two sets:

$$d_{tv}(p, q) = 1/2 \sum_{x \in \Omega} |p_x - q_x| \tag{42}$$

$$= 1/2 \left(\sum_{x \in \Omega^+} (q_x - p_x) + \sum_{x \in \Omega^-} (p_x - q_x) \right), \tag{43}$$

$$\triangleq 1/2 \left(\sum_{x \in \Omega^+} e_x^+ + \sum_{x \in \Omega^-} e_x^- \right). \tag{44}$$

(With $\sum_{x \in \Omega^+} e_x^+ = \sum_{x \in \Omega^-} e_x^-$ since q is a pmf). When we move from the original space to a coarser distribution in $\mathcal{B}$, the total variation is reduced when overestimation errors are grouped with underestimation errors in the same set $\mathcal{A}$ because they cancel each other out:

$$d_{tv}(p^{\mathcal{B}}, q^{\mathcal{B}}) = 1/2 \sum_{\mathcal{A}_i \in \mathcal{B}} \left| \sum_{x \in \Omega^+ \cap \mathcal{A}_i} e_x^+ - \sum_{x \in \Omega^- \cap \mathcal{A}_i} e_x^- \right|, \tag{45}$$

$$= 1/2 \sum_{\mathcal{A}_i \in \mathcal{B}} |E_{\mathcal{A}_i}^+ - E_{\mathcal{A}_i}^-|, \tag{46}$$

where $E_{\mathcal{A}_i}^+ \triangleq \sum_{x \in \Omega^+ \cap \mathcal{A}_i} e_x^+$ and similarly $E_{\mathcal{A}_i}^- \triangleq \sum_{x \in \Omega^- \cap \mathcal{A}_i} e_x^-$. Hence, the gap δ for a given model q_1 between the two total variations can be expressed as

$$\delta_1 \triangleq d_{tv}(p, q_1) - d_{tv}(p^{\mathcal{B}}, q_1^{\mathcal{B}}) = 1/2 \sum_{\mathcal{A}_i \in \mathcal{B}} \min(E_{\mathcal{A}_i}^+, -E_{\mathcal{A}_i}^-) \tag{47}$$

$$(\text{with} \sum_{\mathcal{A}_i \in \mathcal{B}} E_{\mathcal{A}_i}^+ = \sum_{\mathcal{A}_i \in \mathcal{B}} E_{\mathcal{A}_i}^-) \tag{48}$$

Given that $d_{tv}(p, q_1) - d_{tv}(p, q_2) = A \in [0, \infty]$ (q_2 is better), the ranking can be **reversed** if the gap of q_1 for a partitioning $\mathcal{B}$ is greater than the combination of the gap of q_2 and the performance gap A:

$$\delta_1 > \delta_2 + A \tag{49}$$

Table 7: CDM hyperparameters.

Hyperparameters	$K = 6$
hidden size	{ 16, 32, 64}
num. heads	8
depth	2
local size	64
local heads	4
dropout	0.2
T	{ 10, 100, 1000}
batch size	1024
training iterations	1k
learning rate	{7.5e-3, 7.5e-4, 7.5e-5}

Table 8: CNF hyperparameters.

hidden size	{ 16, 32, 64}
num. heads	8
depth	2
local size	64
local heads	4
dropout	0.2
T	{ 10, 100, 1000}
batch size	1024
training iterations	3k
learning rate	{7.5e-3, 7.5e-4, 7.5e-5}

Table 9: ArgmaxAR hyperparameters.

encoder steps	{2, 3 , 4}
encoder bins	{2, 4,5}
context size	{16, 32, 64}
lstm layer	1
lstm size	{16, 32, 64}
context lstm layers	1
context lstm size	{16, 32, 64}
lstm dropout	0.0
batch size	1024
training iterations	2k
learning rate	{7.5e-3, 7.5e-4, 7.5e-5}

This issue can be partially alleviated by generating many random $\mathcal{B}$ and averaging over the results, which we do in our method. Luckily, generating many partitions is not costly at all.

However, in our procedure, not all partitions are drawn randomly. The first $\mathcal{B}^1$ is fixed to Ω_p^Δ (Eqn. (2) in the main text), and then the following $\mathcal{B}^2, \mathcal{B}^3, \dots$ are randomly generated but are still rooted from $\mathcal{B}^1$.

From this, it become apparent that our choice of construction of Ω_p^Δ drives the bias of our procedure. A generative model whose errors (both ϵ^+ and ϵ^-) are evenly distributed across the sets in $\mathcal{S} \in \Omega_p^\Delta$ will have a lower gap $\delta_1 = 1/2 \sum_{\mathcal{S} \in \Omega_p^\Delta} \min(E^+_{\mathcal{A}_i}, -E^-_{\mathcal{A}_i})$. This implies that our procedure is biased towards models that **make mistakes uniformly across high or low pmf values**, and penalizes models for which **the accuracy of q_x correlates with the pmf mass of p_x.**

We believe that it is a reasonable bias to have. Moreover, we validate experimentally that the ordering is preserved.

8 CONFIDENCE INTERVALS OF SYNTHETIC EXPERIMENT

We show visualisation figures of the confidence intervals for the synthetic experiments FLAT and HIGH/LOW in Figure 6.

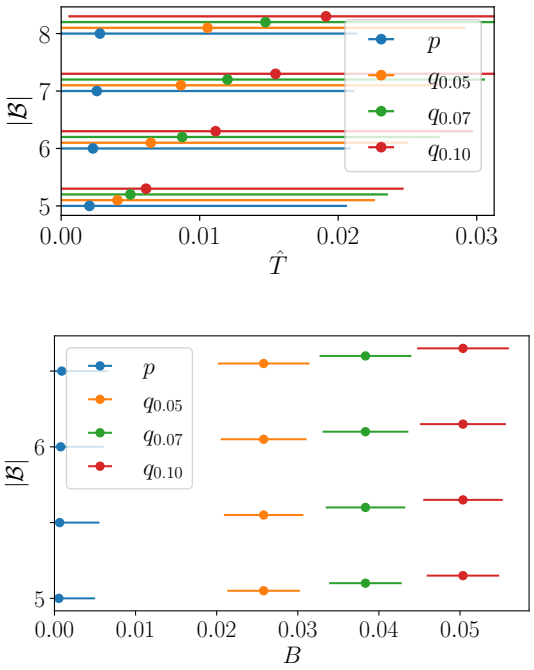
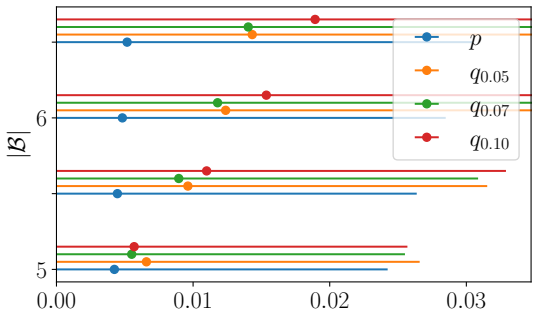


Figure 6: Probabilistic intervals from Theorem 3.1. with $\delta = 10\%$ probability, $m = 100,000$ samples. $|\Omega| = 10^{10}$. FLAT exp. (left), HIGH/LOW exp. (right).

We show the effect of increasing the number of samples on the error intervals on the HIGH/LOW experiments in Figure **??**. As the number of samples increases, the error interval decreases.

9 RELATED WORK - COVERAGE METRICS

To better understand what generative models are learning, Sajjadi et al. (2018) proposed to assess the performance of a generative model through the interpretable notions of precision and recall. Denoting by



$m = 50,000$ samples.

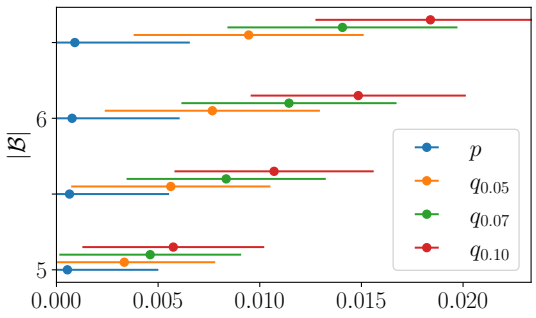


Figure 7: $m = 1M$ samples.

Probabilistic intervals from Theorem 3.1. with $\delta = 10\%$, samples for the HIGH/LOW experiment. With $m = 50,000$ samples (left), the confidence intervals are overlapping. With $m = 1,000,000$ samples (right), some of the confidence intervals do not overlap at higher granularity. For those models and at that granularity, we can state with 80% confidence that the baseline with the lowest total variation estimator $\hat{T}$ has lower total variation than the other baseline at that granularity.

Ω^p and Ω^q the positive supports of p and q, respectively, and defining S as the overlapping support of the two $S = \Omega^p \bigcap \Omega^q$, *precision* measures how much of q is part of p (how small $\Omega^q \setminus S$ is), and *recall* measures how much q misses from p (how small $\Omega^p \setminus S$ is). The mismatch between p and q within the overlap S is viewed as a parameterized combination of both error in precision and error in recall. In practice, the algorithm performs a k-means clustering on the generated samples to estimate the required quantities.

Kynkäänniemi et al. (2019) presented an improved version of the method that better accounts for the concentration of samples and follows more closely the standard definitions of precision and recall. This method still examines the overlap of the spaces and relies on embedding samples into manifolds using a pre-trained classifier network. This introduces biases and many hyperparameters. Moreover, it is computationally intensive as the manifold is built on nearest neighbor principles and is highly sensitive to outliers. These concerns are raised and addressed by Naeem et al. (2020), who introduce *coverage* and *density* metrics, which are based on recall and precision but can be more readily evaluated on manifolds. Naeem et al. (2020) develop an improved method that uses random embeddings to improve the efficiency and employs a more robust method of estimating the probability density to diminish the sensitivity to outliers.

In parallel to the work of Naeem et al. (2020), Djolonga et al. (2020) developed a new metric based on Pareto frontiers of Rényi divergences that encompasses the metrics proposed by Sajjadi et al. (2018) and Kynkäänniemi et al. (2019) but does not rely on data quantization. The theoretical basis of this work was further extended by Liu et al. (2021), who assessed the sample complexity of the evaluation method.

More recently,the work by (Alaa et al., 2022) addresses the issue of detecting samples that are anomalously close to the training set. This was a shortcoming of the prior methods. Robustness to repetition or near-repetition of the training set is critical for any sample-based metric (Theis et al., 2016). The metric proposed by Alaa et al. (2022) is a 3-dimensional evaluation metric based on the quality of the samples, the coverage of the distribution, and the generalization capability. These three aspects are also addressed by recent approaches based on density estimation (Jiralerspong et al., 2023). Lastly, (Abdar et al., 2021) have recently raised concerns about the robustness of precision and recall metrics in high-dimensional settings and proposed a solution to mitigate this issue.