# MIND: Monge Inception Distance for Generative Models Evaluation

**Quentin Berthet**[1] **Yu-Han Wu**[1,2] **Clément Crepy**[1]
**Romuald Elie**[1] **Klaus Greff**[1] **Michaël E. Sander**[1]

[1]Google DeepMind [2] LPSM, Sorbonne Université

## Abstract

We propose the Monge Inception Distance (MIND), a metric for evaluating generative models that addresses key limitations of the widely adopted Fréchet Inception Distance (FID). The MIND metric leverages the sliced Wasserstein distance to compare distributions by averaging one-dimensional optimal transport distances, efficiently computed via sorting. This approach circumvents the estimation of high-dimensional means and covariance matrices, which underlie FID's poor sample complexity and vulnerability to adversarial attacks. We empirically demonstrate three primary advantages: (i) it is more sample-efficient by one order of magnitude, (ii) it is faster to compute by two orders of magnitude, (iii) it is more robust to adversarial attacks such as moment-matching. We show that MIND with 5k samples can replace the evaluation performance of FID with 50k samples, providing high correlation with this standard benchmark and superior discriminative performance. We further demonstrate that even smaller sample sizes (e.g., 1k or 2k) remain highly informative for rapid model iteration.

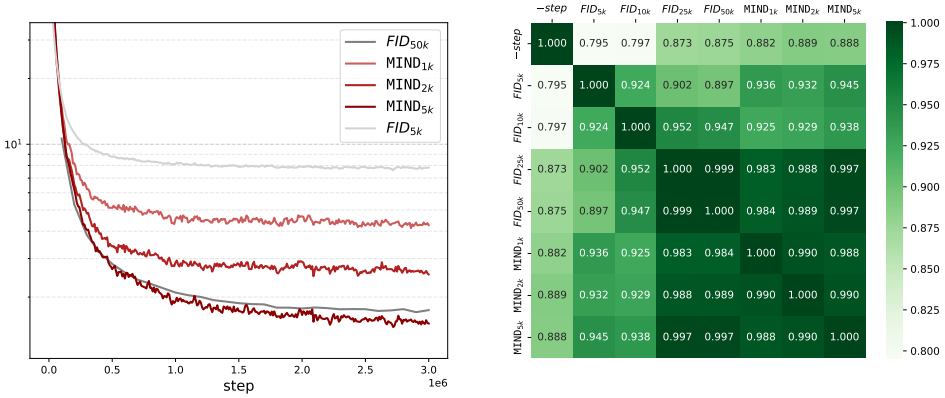



Figure 1: (**Left**) MIND metric during a diffusion model training run on ImageNet-64 (log scale), illustrating how $\text{MIND}_{5k}$ can be used to replace $\text{FID}_{50k}$, with a larger range - see Section 4.3. (**Right**) Correlation with number of training steps - better for $\text{MIND}_{1k}$ and $\text{MIND}_{5k}$ than FID with 50k samples.

## 1 Introduction

Generative models, especially diffusion models (Ho et al., 2020), have set new standards in high-quality data synthesis. This progress has spurred innovation across numerous fields, from creative

arts to scientific simulation. However, as models grow in complexity, the metrics used to evaluate them have struggled to keep pace (Stein et al., 2023). The de facto standard, the Fréchet Inception Distance (FID) (Heusel et al., 2017) is based on a Gaussian approximation of pre-trained network embeddings, such as the Inception-v3 model (Salimans et al., 2016).

This metric relies on estimating high-dimensional mean and covariance matrices from inception embeddings. However, this sample-heavy approach typically requires 50k samples (Chong and Forsyth, 2020), creating a significant development bottleneck. Furthermore, because FID only considers the first two moments of the distributions, it is not a true distance metric and is vulnerable to adversarial "hacking" without corresponding visual improvements (Sajjadi et al., 2018).

In this work, we introduce the Monge Inception Distance (`MIND`) addressing these limitations. Based on optimal transport theory and named in honor of Gaspard Monge, who introduced the optimal transport problem (Monge, 1781), `MIND` leverages the sliced Wasserstein distance (Rabin et al., 2011). Instead of relying on Gaussian simplification as in FID, `MIND` reduces the complexity of high-dimensional optimal transport comparison by averaging many one-dimensional projections, where the transport problem is solved exactly via a simple, parallelizable sorting operation. This captures finer distributional details without the statistical instability inherent in high-dimensional matrix estimation - see (Villani, 2008; Peyré and Cuturi, 2019) for a modern perspective on optimal transport theory and applications.

We highlight that this approach yields stable, high-quality evaluation using only 5k samples – an order of magnitude fewer than FID – while offering $100\times$ faster computation and increased robustness to adversarial moment-matching. We statistically validate the performance of `MIND` across various hypothesis testing problems at different sample sizes. Finally, while we present our results using Inception-v3 embeddings to facilitate a direct comparison with the current FID benchmark, the `MIND` metric is fundamentally embedding-agnostic. It is modality-independent and can be seamlessly applied to any representation space, including CLIP (Radford et al., 2021), DINO (Oquab et al., 2024; Siméoni et al., 2025), or specialized embeddings for audio and video synthesis.

**Main contributions.** In this work, we introduce `MIND`, a metric for improved evaluation of generative models. We demonstrate the following advantages of this metric:

- **Sample Efficiency**: We show that $\texttt{MIND}_{5k}$ provides a stable evaluation that correlates highly with $\text{FID}_{50k}$, enabling reliable benchmarking with $10\times$ fewer samples.
- **Computational Speed and Memory Efficiency**: Due to its reliance on 1D sorting rather than high-dimensional matrix operations, `MIND` is over $100\times$ faster to compute, and requires $10\times$ less memory, facilitating real-time evaluation during training.
- **Metric Robustness**: Since `MIND` is derived from a proper distance, we show that it is significantly more resistant to "metric hacking" via moment-matching attacks that can artificially lower FID.
- **Discriminative Power**: Our experiments show that `MIND` more reliably distinguishes between model checkpoints and identifies subtle image perturbations at low sample sizes.

## 2 Generative model evaluations

We consider the problem of evaluating a generative model $g_\theta$, that generates outputs by mapping noise $Z \sim \mathcal{N}(0, I)$ to data outputs (e.g. images) $a = g_\theta(Z)$. In standard practice, these outputs are passed through a pre-trained feature extraction model $\psi_w$ to obtain embeddings. For an Inception-v3 model (Szegedy et al., 2016), these embeddings typically reside in dimension $d = 2,048$.

The performance of the model is measured by the statistical distance between the distribution of generated embeddings, $X = \psi_w(g_\theta(Z)) \sim p_\theta$, and the distribution of real dataset embeddings, $Y = \psi_w(D) \sim p_{data}$. In practice, this distance is estimated using finite samples of size $n$, denoted as the empirical distributions $\hat{p}_{n,\theta}$ and $\hat{p}_{n,data}$ (see Appendix A). Any measure of statistical distance between these distributions can be used, and we consider in this work several inception distances, defined as follows for any distance or divergence $\Delta$ between distributions (see, e.g. Cover, 1999).

**Definition 2.1** (General - Inception distance)**.** *Let $X = \psi_w(g_\theta(Z)) \sim p_\theta$, $Y = \psi_w(D) \sim p_{data}$. For a distribution distance function $\Delta$, the performance of the model $g_\theta$ is given by $\Delta ID(p_\theta, p_{data})$. With a sample of size $n$, its empirical estimate is the plug-in value $\Delta ID(\hat{p}_{n,\theta}, \hat{p}_{n,data})$.*

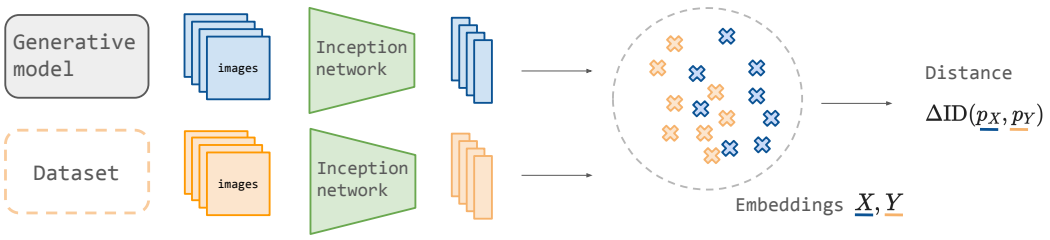



Figure 2: General pipeline for evaluating generative model sampling distance to a dataset.

### 2.1 Existing method: Fréchet Inception Distance - FID

FID is the most widely adopted instance of an inception distance. It measures the distance between two distributions based on their first two moments: the mean ($\mu$) and covariance ($\Sigma$), using the squared 2-Wasserstein distance $W_2^2$ (see Appendix A.2)

**Definition 2.2** (Fréchet Inception Distance - FID)**.** *Let $X = \psi_w(g_\theta(Z)) \sim p_\theta$ and $Y = \psi_w(D) \sim p_{data}$ and $\mu_X, \Sigma_X$ and $\mu_Y, \Sigma_Y$ be the means and covariances of $p_\theta$ and $p_{data}$, respectively. The FID is defined as the squared 2-Wasserstein distance between two fitted Gaussians:*

$$FID(p_\theta, p_{data}) = \|\mu_X - \mu_Y\|^2 + \text{tr}(\Sigma_X + \Sigma_Y - 2(\Sigma_Y\Sigma_X)^{1/2})\,.$$

*In practice, this is computed using empirical sample means and covariances $\hat{\mu}_n$ and $\hat{\Sigma}_n$.*

This approach was originally motivated as a way to bypass the high computational and statistical complexity of a direct, sample-based Wasserstein distance by using a Gaussian approximation.

**Drawbacks** Despite its widespread use and role as de facto standard metric, there are several drawbacks in using this distance, as noted in several works (see, e.g. Karras et al., 2017; Stein et al., 2023; Jayasumana et al., 2024; Bischoff et al., 2024; Yang et al., 2026)

- Computing this distance is based on the estimate $\hat{\Sigma}_n$ of a $d$-by-$d$ covariance. It is rank-deficient for $n \leq d$, creating numerical and statistical issues when estimating the second term, unless the sample size is at least of order $d$, which is 2048 for inception networks. For images, this means that the sample sizes usually used are 10k or 50k (Bińkowski et al., 2018; Chong and Forsyth, 2020), with a high impact on evaluation time and cost.
- FID is also not a proper distance (Jayasumana et al., 2024): two distributions can have the same mean and covariance and be very different (see, for example, Billingsley, 2017, Section 30). We show that, as a consequence, it is not robust, and this fact can be leveraged to artificially reduce the FID without visually altering the images (see Section 4.5).

We also evaluate other inception-based distances proposed as metrics, as means of comparison, such as the *Maximum mean discrepancy* (MMD), or Sinkhorn divergence. We provide full definitions and a discussion in appendix (Section A.3), and include them in comparisons.

## 3 Proposed method: Monge Inception Distance - `MIND`

We propose the following metric to overcome these challenges, based on the sliced Wasserstein distance which has several known advantages, both in terms of statistical and computational complexity. It is an average of the Wasserstein distances of the distribution of projections, over all unit directions (see, e.g. Rabin et al., 2011; Nadjahi, 2021, and Appendix A.2 in this work for details).

The sliced Wasserstein distance is a proper distance - it is equal to 0 if and only if both distributions are equal (Bonnotte, 2013, Proposition 5.1.2). We use this approach for our `MIND` metric, taking the average of Wasserstein distances projected along finitely many unit directions for an estimate.

**Definition 3.1** (Monge Inception Distance)**.** *Let $X = \psi_w(g_\theta(Z)) \sim p_\theta$ and $Y = \psi_w(D) \sim p_{data}$, and $\mathcal{U}(S)$ be the uniform distribution on the unit sphere. `MIND` is given by averaging $W_2^2$ distances for projections of the distributions along unit directions, with a multiplicative scaling $\alpha = 3d$*

$$MIND(p_\theta, p_{data}) = \alpha \mathbb{E}_{u\sim\mathcal{U}(S)}[W_2^2(u^\intercal p_\theta, u^\intercal p_{data})]\,,$$

*where $u^\intercal p_\theta$ (resp. $u^\intercal p_{data}$) is the distribution of $u^\intercal X$ when $X \sim p_\theta$ (resp. of $u^\intercal Y$ when $Y \sim p_{data}$) and $d$ is the data dimension.*

*For finite samples $(X_j)_{j\in[n]}$, $(Y_j)_{j\in[n]}$, random unit directions $(u_i)_{i\in[M]}$ and $\alpha = 3d$, it is given by*

$$\mathit{MIND}(\hat{p}_{n,\theta}, \hat{p}_{n,data}) = \frac{\alpha}{M}\sum_{i=1}^{M} W_2^2(u_i^\intercal \hat{p}_{n,\theta}, u_i^\intercal \hat{p}_{n,data}) = \frac{\alpha}{nM}\sum_{i=1}^{M}\sum_{j=1}^{n} |\mathit{sort}(u_i^\intercal X)_j - \mathit{sort}(u_i^\intercal Y)_j|^2 .$$

**Remarks**

- `MIND` relies on two finite sample estimates: the 1D Wasserstein distance over samples $X_j$ and $Y_j$ of size $n$, and the expectation $\mathbb{E}_{u\sim\mathcal{U}(S)}$ over $M$ random unit directions $u_i$.
- Although we adopt the name of Inception Distance for consistency with established literature, the formulation of `MIND` does not depend on the Inception architecture. The mapping function $\psi_w$ can represent any feature extractor; consequently, `MIND` serves as a general-purpose tool for evaluating distributional similarity across diverse data modalities and embedding models.
- This 1D formulation allows a more stable evaluation using an order of magnitude fewer samples – with $n$ of order 5k rather than 50k (see Sections 4.3 and 4.4). Furthermore, because the sliced Wasserstein distance is a proper distance, having matching means and covariance matrices is not sufficient for `MIND` to be zero, making it inherently robust to moment-matching hacking (see Section 4.5).

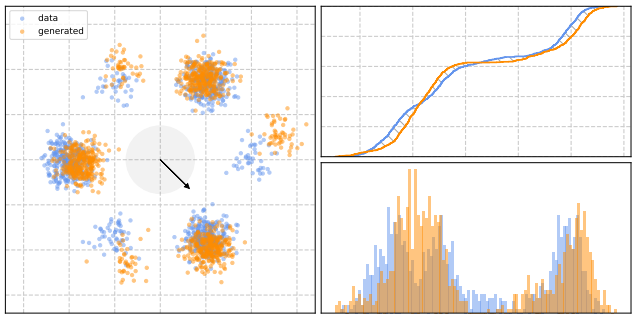



Figure 3: Computation of `MIND` based on the idea of Sliced Wasserstein, illustrated in 2D with a single projection. (**Left**) Two samples of synthetic embeddings (orange and blue), along with the unit sphere and a random unit direction $u$. (**Bottom Right**) The two histograms of distributions of the projections along $u$. (**Top Right**) The associated cumulative distribution functions (cdf), the hatched area is related to 1D Wasserstein distances along $u$: it is the $W_1$ distance, and used for all convex costs with pairwise sorted distances.

- Leveraging the exact solution of 1D transport problem (see, e.g. Peyré and Cuturi, 2019, and Appendix A.2 in this work), the distance simplifies to pair-wise difference between sorted elements:

$$W_2^2(\hat{p}_n, \hat{q}_n) = \frac{1}{n}\sum_{j=1}^{n} |\mathrm{sort}(x)_j - \mathrm{sort}(y)_j|^2 ,$$

  where $\mathrm{sort} : \mathbb{R}^n \to \mathbb{R}^n$ is the function that maps a vector $x \in \mathbb{R}^n$ to its copy sorted in nondecreasing order, i.e. $\mathrm{sort}(x) = (x_{\sigma(1)}, \ldots, x_{\sigma(n)})^\intercal$ such that $x_{\sigma(1)} \leq \ldots \leq x_{\sigma(n)}$ (ties do not make this function ambiguous). Since the sorting operation runs in $O(n \log n)$ time, `MIND` avoids the need to estimate or store high-dimensional objects (e.g. $d \times d$ matrices for FID, $n \times n$ matrices for other distances). Similarly to FID, we use the square of the distance rather than taking a square root of the average. We also use a multiplicative scaling factor $\alpha$—see discussion in Section 4.2.

# 4 Experiments

## 4.1 Implementation

As noted above, estimating `MIND` on two samples of size $n$ is both computationally and conceptually easy. It requires only trivially parallelizable projections and sorting operations. As such, it is particularly adapted to modern accelerated-oriented hardware and software. We provide in Figure 4 both a JAX (Bradbury et al., 2018) and PyTorch (Paszke et al., 2019) implementation, in the form of a short code snippet that can be directly used in an evaluation pipeline. We also provide in Section 4.7 and 4.8 experimental results showcasing the computation time and memory advantages of this algorithm compared to other methods.

(a) JAX

```jax
def monge_inception_distance(
  x, y, rng_seed, n_projections=100
):
  """MIND metric.

  Args:
    x: Input generated features.
    y: Ground truth features.
    rng_seed: An integer for the seed of RNG.
    n_projections: Number of projections to use.

  Returns:
    The value of the MIND metric.
  """
  num_samples, d = x.shape
  assert num_samples == y.shape[0]
  ALPHA = 3 * d
  key = jax.random.PRNGKey(rng_seed)
  u_proj = jax.random.normal(key, (n_projections, d))
  u_proj /= jnp.linalg.norm(u_proj, axis=-1, keepdims=True)

  x_proj = u_proj @ x.T
  y_proj = u_proj @ y.T
  dists = jnp.mean((
    jax.lax.top_k(x_proj, num_samples)[0]
    - jax.lax.top_k(y_proj, num_samples)[0]
  ) ** 2, axis=1)

  return ALPHA * jnp.mean(dists)
```

(b) PyTorch

```pytorch
def monge_inception_distance_torch(
  x, y, rng_seed, n_projections=100
):
  """MIND metric.

  Args:
    x: Input generated features.
    y: Ground truth features.
    rng_seed: An integer for the seed of RNG.
    n_projections: Number of projections to use.

  Returns:
    The value of the MIND metric.

  """
  num_samples, d = x.shape
  assert num_samples == y.shape[0]

  ALPHA = 3 * d
  generator = torch.Generator(device=x.device).manual_seed(
    rng_seed
  )
  u_proj = torch.randn(
    (n_projections, d),
    generator=generator,
    dtype=x.dtype,
    device=x.device
  )
  u_proj /= torch.linalg.norm(u_proj, dim=-1, keepdim=True)

  x_proj = u_proj @ x.T
  y_proj = u_proj @ y.T
  dists = torch.mean((
    torch.topk(x_proj, num_samples, dim=-1).values
    - torch.topk(y_proj, num_samples, dim=-1).values
  ) ** 2, dim=1)

  return ALPHA * torch.mean(dists)
```

Figure 4: JAX (a) and PyTorch (b) implementation of MIND.

### 4.2 Hyperparameter choices

As noted in Definition 3.1, we scale the MIND metric by a multiplicative factor $\alpha > 0$. This is done so that the order of magnitude of this metric matches those of FID. This proximity helps to compare values of MIND to those of FID, and is chosen to favor adoption. Based on an analysis on ImageNet-64, we have found that taking $\alpha = 3 \times d \approx 6,000$ is a good fit, especially later in a training run (where $d = 2,048$ is the dimension of the embedding space) - see Figure 5. We also observe that $\text{MIND}_{5k}$ has more range than FID for any sample size, with higher values early in the run (above even those of $\text{FID}_{5k}$), and aligned with $\text{FID}_{50k}$ later in the run, and better aligned with the number of steps in a training run (see Figure 1, right). We also compute MIND and FID score with features of various dimension. The features are obtained by truncating the Inception-v3 features to the target dimension. Figure 5 (Right) shows that the MIND remains an affine relation with respect to FID while varying the dimension of the feature space in the log-log plot—justifying the choice of the scaling factor $\alpha \propto d$.

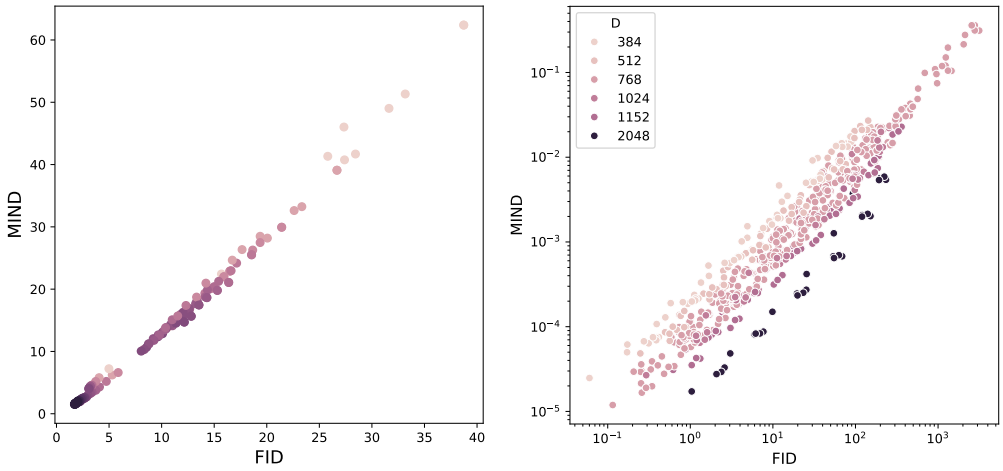



Figure 5: $\text{MIND}_{5k}$ and $\text{FID}_{50k}$ (and other sample sizes) for a model at different steps of training on ImageNet-64. **Left**: All MIND metrics are rescaled by a factor $\alpha \approx 6,000$ chosen to optimize proximity with FID, colors indicating the step at which the metric is evaluated. **Right**: MIND without scaling factor for various embedding dimension, colors indicating the dimension in which the metric is evaluated.

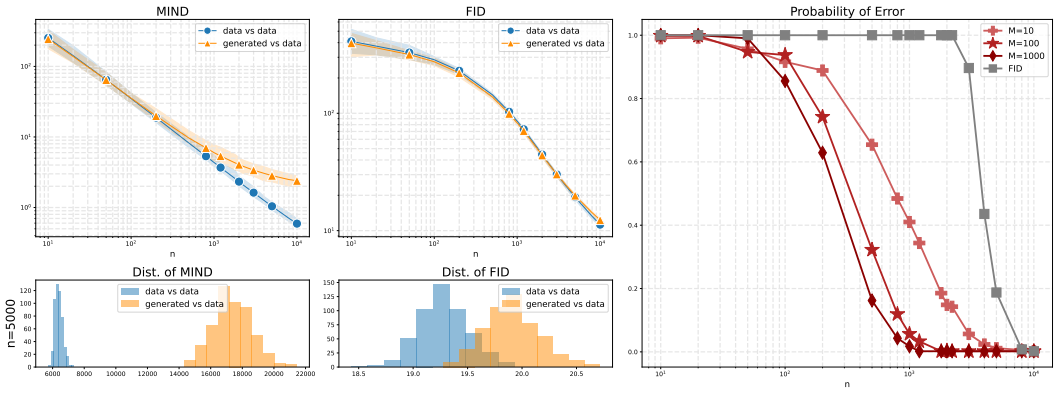



Figure 6: (**Top Left and Middle**) Behavior of the MIND and FID metric in $n$, to distinguish true images from the dataset (base - in blue) from generated images (model - in orange). (**Bottom Left and Middle**) Histogram of the trials for $n = 5,000$ - A bigger gap is better. (**Right**) Probability of error defined in Section 4.4.1 for three values of $M \in \{10, 100, 1000\}$.

### 4.3 MIND analysis

We illustrate the behavior of the MIND metric by analyzing its dependency on $n$ (the number of samples from the distributions) and $M$ (the number of random projections). Fixing a number $k > 0$ (varies in different settings), we evaluate its ability to correctly order $k$ different distributions $p^1, \dots p^k$ relative to the true data distribution. Specifically, we calculate the probability of error in correctly ranking the sequence of distances $\texttt{MIND}(\hat{p}_n^1, \hat{p}_{n,\text{data}}), \dots, \texttt{MIND}(\hat{p}_n^k, \hat{p}_{n,\text{data}})$. This measures the ability of the metric to distinguish different images from the elements of the dataset. Our observations indicate that $n = 5,000$ samples is sufficient to reliably distinguish these distributions (we benchmark this against other metrics quantitatively in Section 4.4).

We also plot the dependency of the estimated metric on $M$, the number of uniformly chosen random projections. This dependency is easier to analyze, since the Monte-Carlo estimate is obtained by averaging unbiased terms to compute the expected Wasserstein distance. We observe that choosing $M$ in the range $[100,\ 1000]$ is sufficient (see Appendix C).

### 4.4 Metric comparison

Running evaluations during training of a diffusion model, we observe that instead of using $\text{FID}_{50k}$ (commonly used post-training because of the cost and time associated with the high sample size), we can use $\texttt{MIND}_{5k}$ (we evaluate their precisions more quantitatively in the rest of this section). As visible in Figure 5, these two metrics are highly correlated, especially later during training.

In order to compare different metrics in a principled fashion, we evaluate how useful they are to distinguish distributions. This provides natural *tasks* for which these metrics can be evaluated as *methods*, through the lens of statistical hypothesis testing. This can also be understood as a comparison in distribution of the metrics (for random samples), rather than a single value. Given sample size $n$ and metric $\Delta$, we state our three statistical hypothesis testings in the following.

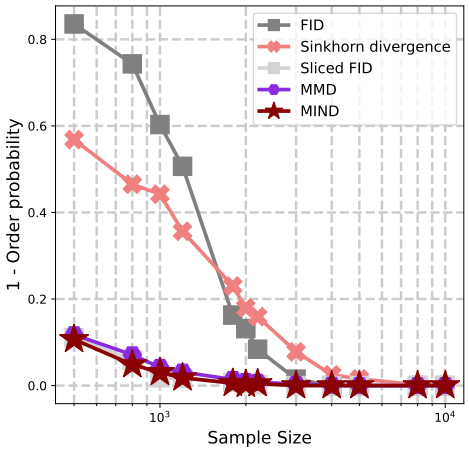



Figure 7: Sample complexity measured by the probability of error for the correct order at five different steps of training.

#### 4.4.1 Generated vs. true data

This is done by comparing two distributions $p_\theta$ (for some pre-trained model $g_\theta$, with parameters $\theta$) with $p_{\text{data}}$, and estimating $\Delta(\hat{p}_{n,\theta}, \hat{p}_{n,\text{data}})$. Our diffusion model utilizes a U-Net backbone (Ronneberger et al., 2015; Nichol and Dhariwal, 2021) trained on Imagenet-64. Experiments relying on real data use a fixed set of 100,000 original Imagenet-64 images. Conversely, for evaluations involving data generated from models, we generate a dedicated fixed set of 50,000 samples from each model under evaluation.

We are comparing the values of the metric under two settings, where $\hat{p}_{n,\theta}$ is a distribution of $n$ samples generated from a trained model, and $\hat{p}_{n,\text{data}}$ and $\hat{p}'_{n,\text{data}}$ are two independent samples of size $n$ from the data. The probability of error is defined as:

$$\mathbf{P}\big(\Delta(\hat{p}_{n,\text{data}}, \hat{p}'_{n,\text{data}}) \geq \Delta(\hat{p}_{n,\text{data}}, \hat{p}_{n,\theta})\big)\,.$$

This measures the ability of the metric to distinguish generated images from elements of the dataset. The results are given in Figure 6. We remark that, MIND is able to separate the two distributions as soon as $n \geq 5,000$ (Figure 6 Left column), while FID requires more than 10k samples to do so (Figure 6 middle column).

#### 4.4.2 Monotonicity

We compare MIND with other metrics using a diffusion model $g_\theta$ trained on the ImageNet-64 dataset (Deng et al., 2009), from which we selected five models, $g_{\theta_1}, \ldots, g_{\theta_5}$, corresponding to five distinct training checkpoints and generate 50k images with each of them. The probability of error in ranking these checkpoints correctly is:

$$1 - \mathbf{P}\big(\Delta(\hat{p}_{n,\text{data}}, \hat{p}_{n,\theta_1}) \geq \ldots \geq \Delta(\hat{p}_{n,\text{data}}, \hat{p}_{n,\theta_k})\big)\,.$$

For several sample sizes $n$ ranging from 10 to 10k, we perform 512 independent trials for each metric. We observe that MIND, MMD and sliced FID achieves similar performance in this test (Figure 7).

#### 4.4.3 Perturbations

We consider three different types of image perturbations, the severity of perturbation is given by a parameter $\varepsilon$. Each experiment is performed with 512 independent trials. We measure the performance of each metric using

$$1 - \mathbf{P}\big(\Delta(\hat{p}_{n,\text{data}}, \hat{p}_{n,\text{data},\varepsilon_1}) \leq \ldots \leq \Delta(\hat{p}_{n,\text{data}}, \hat{p}_{n,\text{data},\varepsilon_k})\big)\,.$$

which is the probability of failing to order all perturbation levels. The results are summarized in Figure 8. We highlight that, for $n \geq 5,000$, MIND achieves the same level of performance as MMD while FID is worse on all tasks.

**Gaussian blur.** We select a perturbation level $\varepsilon \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. We apply a Gaussian filter with standard deviation $\varepsilon$ to each image in ImageNet-64.

**Rectangle.** We select a perturbation level $\varepsilon \in \{0.05, 0.1, 0.15, 0.2\}$. We randomly place 5 squares of size $10 \times 10$ with $\varepsilon$ opacity in each image of ImageNet-64.

**Mixture of datasets.** We select a perturbation level $\varepsilon \in \{1\%, 3\%, 5\%, 7\%, 10\%\}$. We draw samples with proportion of $(1 - \varepsilon)$ from ImageNet-64 and of $\varepsilon$ from CelebA (Liu et al., 2015).

### 4.5 Robustness to metric hacking with moment matching

As mentioned above, one of the weaknesses of FID is that it is not a proper distance. Indeed, since this metric is only a function of the means and covariances of the considered distributions, if $p$ and $q$ share the same first and second moments, then $\text{FID}(p, q) = 0$. This is not the case for proper metrics derving from distances such as MIND. We leverage this fact to create an artificial distribution of samples that have a desired mean and covariance. This construction is based on the following property whose proof is in Appendix B.1.

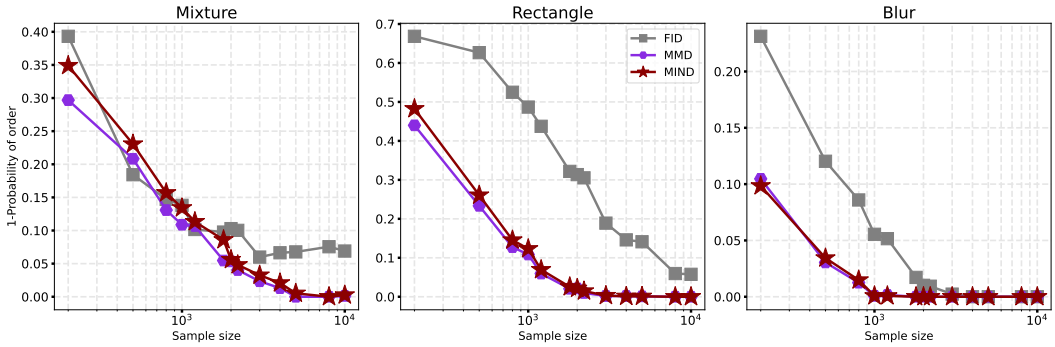



Figure 8: Sample complexity measured by the probability of detecting a small perturbation.

**Proposition 4.1.** *Let $p$ be a target distribution over $\mathbb{R}^d$ with mean $\mu$ and covariance $\Sigma$, whose eigendecomposition is*

$$\Sigma = USU^\intercal = \sum_{i=1}^{r} \lambda_i u_i u_i^\intercal \,.$$

*Define the $2r$ vectors $v_i^{(+)}$ and $v_i^{(-)}$ indexed by $i \in [r]$, by*

$$v_i^{(+)} = \mu + \alpha u_i \,, \quad v_i^{(-)} = \mu - \alpha u_i \,,$$

*with $\alpha = \sqrt{\boldsymbol{Tr}(\Sigma)}$. Define $\pi_i^{(+)} = \pi_i^{(-)} = \lambda_i / (2\boldsymbol{Tr}(\Sigma))$ and note that the $\pi_i$ are nonnegative and sum to 1. Let $\hat{q}$ be the distribution of the $v_i$'s, each with probability $\pi_i$, given by*

$$\hat{q} = \sum_{i=1}^{r} \pi_i^{(+)} \delta_{v_i^{(+)}} + \sum_{i=1}^{r} \pi_i^{(-)} \delta_{v_i^{(-)}} \,.$$

*It holds that*

$$\mathbb{E}_{\hat{q}}[v] = \mu \,, \quad \mathbb{E}_{\hat{q}}[(v-\mu)(v-\mu)^\intercal] = \Sigma \,, \quad \textit{FID}(\hat{q}, p) = 0 \,.$$

### 4.6 Moment matching procedure

This proposition can be leveraged to perform *metric hacking* with moment matching: For a batch of $n$ images $a^0$ with embedding distribution $\hat{q}^0$ we construct $a = a^0 + \varepsilon$ whose embeddings have distribution $\hat{q}$ such that the metric $\Delta(\hat{q}, \hat{p}_{n,\text{data}})$ is much smaller than $\Delta(\hat{q}^0, \hat{p}_{n,\text{data}})$ (for some data distribution $p_{\text{data}}$) by optimizing the moments of these embeddings, using the target vectors given by Proposition 4.1 with no visually discernible alteration (see Figure 11 in Appendix C.4).

We do so in the following manner: for $n = 2r$ and a batch $a^0 \in \mathbb{R}^{2r \times [\text{dims}]}$ of $2r$ images, each of shape $[\text{dims}]$ (e.g. $[512, 512, 3]$), and a target distribution $p_{\text{data}}$ over $\mathbb{R}^d$, we consider the following objective, aiming to give each $a_i$ an embedding close to the target $v_i$

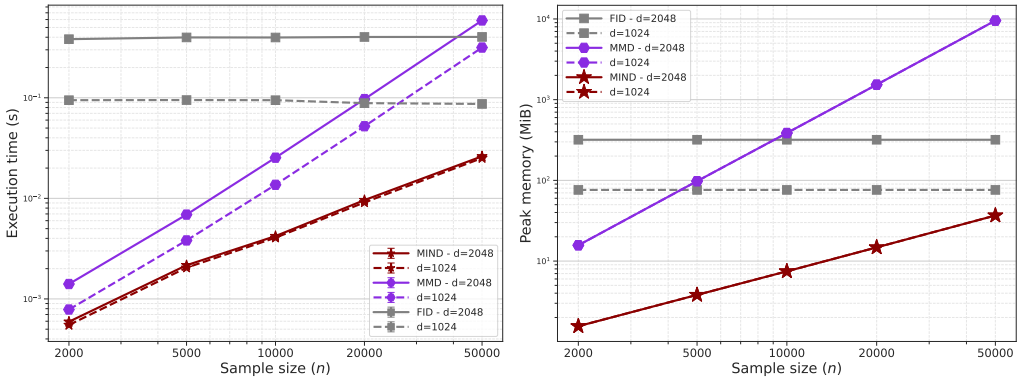
$$\min_{a \in \mathbb{R}^{2r \times [\text{dims}]}} \ell(a) = \min_{a \in \mathbb{R}^{2r \times [\text{dims}]}} \sum_{i=1}^{2r} \|\psi_w(a_i) - v_i\|^2 \,.$$

We initialize $a$ at $a^0$ that is $2r$ copies of the same image. This optimization problem is highly parallelizable since the loss is fully separable over each of the $a_i$, and we can use stochastic based optimization methods to solve it. If the batch $a$ satisfy $\ell(a) = 0$, then the FID of the distribution of the $a_i$ with probabilities $\pi_i$ is also 0, and we show that optimizing this loss reduces the FID significantly.

In this experiment, we use a full-rank batch, $r = 2048$, the dimension of the latent space. Therefore, the total batch size is

| **Metric** | ratio |
|---|---|
| **FID** | 11.2% |
| $\mu$**FID** | 2.6% |
| $\sigma$**FID** | 4.2% |
| **MMD** | 12.2% |
| MIND | 31.1% |

Table 1: Robustness of several metrics under moment matching




(a) Execution time

(b) Peak Memory

Figure 9: Walltime and peak memory comparison for MIND, MMD, and FID

$n = 4096$ and we separate the optimization problem. We use in our evaluation $M = 1000$ for MIND and 50k to compute the reference mean and covariance for the FID. The results summarized in Table 1 show that several of these metrics are highly sensitive to moment matching hacking, with only $10\%$ or less of the metric remaining for the baseline metrics (and much less for sample-efficient metric versions of the FID), and that while affected, the MIND metric is much more robust.

### 4.7 Computation time comparison

We compare the running time of *computing* the different metrics on TPUv4, given two sets of embeddings with sample size $n$. We emphasize that this is only the time to compute the metric, not to generate the samples, which is roughly linear in $n$. In Figure 9a, we observe that computing MIND at its recommended sample size of 5k is more than 2 orders of magnitude faster than computing FID. This is an additional difference, on top of the time necessary to sample a much larger sample when using FID.

### 4.8 Peak memory comparison

We compare the peak memory required to compute the different metrics on a TPUv4, using two sets of embeddings with sample size $n$. Note that these measurements reflect only the additional memory consumed during metric computation which do not include the memory occupied by the input data itself, the latter results are provided in Appendix C.2. In Figure 9b, we highlight that computing MIND at its recommended sample size ($n = 5$k) requires over an order of magnitude less memory than computing either MMD or FID. (Note that the curves for MMD and MIND collapse across different input dimensions, resulting in overlapping lines).

## Conclusion

In this work, we introduced MIND, a metric for evaluating generative models that addresses statistical and computational limitations of FID. Our empirical results demonstrate that MIND is faster to compute and achieves stable evaluations with sample sizes as low as 2k, compared to the 50k typically required for FID. Furthermore, as a proper distance metric, MIND exhibits better robustness to moment-matching adversarial attacks than other metrics, while being affected by it. As a purely statistical metric, MIND measures the distributional distance to a reference dataset (such as the training data). It is not designed to evaluate other qualitative aspects of the generated images, such as visual aesthetics or text legibility. We believe this metric provides a rigorous, efficient, and reliable standard for assessing the quality of modern generative models.

## References


Billingsley, P. (2017). *Probability and measure*. John Wiley & Sons.

Bischoff, S., Darcher, A., Deistler, M., Gao, R., Gerken, F., Gloeckler, M., Haxel, L., Kapoor, J., Lappalainen, J. K., Macke, J. H., et al. (2024). A practical guide to sample-based statistical distances for evaluating generative models in science. *arXiv:2403.12636*.

Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. (2018). Demystifying MMD GANs. In *The Sixth International Conference on Learning Representations*.

Bonnotte, N. (2013). *Unidimensional and evolution methods for optimal transportation*. PhD thesis, Université Paris Sud-Paris XI; Scuola normale superiore (Pise, Italie).

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., et al. (2018). JAX: composable transformations of Python+ NumPy programs.

Chong, M. J. and Forsyth, D. (2020). Effectively unbiased FID and inception score and where to find them. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6070–6079.

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database . In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pages 248–255, Los Alamitos, CA, USA. IEEE Computer Society.

Dudley, R. M. (1969). The speed of mean Glivenko-Cantelli convergence. *Annals of Mathematical Statistics*, 40:40–50.

Genevay, A., Peyre, G., and Cuturi, M. (2018). Learning generative models with Sinkhorn divergences. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1608–1617. PMLR.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773.

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 Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.

Jayasumana, S., Ramalingam, S., Veit, A., Glasner, D., Chakrabarti, A., and Kumar, S. (2024). Rethinking FID: Towards a better evaluation metric for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9307–9315.

Kantorovich, L. V. (1942). On the translocation of masses. *Doklady Akademii Nauk SSSR*, 37(7-8):227–229.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). Progressive growing of GANs for improved quality, stability, and variation. *arXiv:1710.10196*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738.

Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences*, pages 666–704.

Nadjahi, K. (2021). *Sliced-Wasserstein distance for large-scale machine learning : theory, methodology and extensions*. Theses, Institut Polytechnique de Paris.

Nadjahi, K., Durmus, A., Chizat, L., Kolouri, S., Shahrampour, S., and Simsekli, U. (2020). Statistical and topological properties of sliced probability divergences. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20802–20812. Curran Associates, Inc.

Nichol, A. Q. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. (2024). DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Peyré, G. and Cuturi, M. (2019). Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Rabin, J., Peyré, G., Delon, J., and Bernot, M. (2011). Wasserstein barycenter and its application to texture mixing. In *International conference on scale space and variational methods in computer vision*, pages 435–446. Springer.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Sajjadi, M. S. M., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. (2018). Assessing generative models via precision and recall. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training GANs. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Siméoni, O., Vo, H. V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al. (2025). DINOv3. *arXiv:2508.10104*.

Stein, G., Cresswell, J., Hosseinzadeh, R., Sui, Y., Ross, B., Villecroze, V., Liu, Z., Caterini, A. L., Taylor, E., and Loaiza-Ganem, G. (2023). Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 3732–3784. Curran Associates, Inc.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826. IEEE.

Villani, C. (2008). *Optimal transport: old and new*, volume 338. Springer.

Yang, J., Geng, Z., Ju, X., Tian, Y., and Wang, Y. (2026). Representation Fréchet loss for visual generation. *arXiv:2604.28190*.

# A Definitions

## A.1 Empirical measures

**Definition A.1.** *For a sample $Y_1, \dots, Y_n$ of size $n$ from some data distribution $p_{data}$, we denote by $\hat{p}_{n,data}$ the* empirical distribution *of the $Y_i$s, defined by*

$$\hat{p}_{n,data} = \frac{1}{n} \sum_j \delta_{Y_j} .$$

*Similarly, for $X_1, \dots, X_n$ from $p_\theta$ we denote by $\hat{p}_{n,\theta}$ the empirical distribution of the $X_i$s*

$$\hat{p}_{n,\theta} = \frac{1}{n} \sum_j \delta_{X_j} .$$

## A.2 Optimal transport

**Definition A.2.** *The optimal transport problem for Euclidean cost, also called the 2-Wasserstein distance is defined for two probability distributions $p, q \in \mathcal{P}_2(\mathbb{R}^d)$ with finite second moments as*

$$W_2^2(p,q) = \min_{T : T_\# p = q} \mathbb{E}_{X \sim p}[\|X - T(X)\|^2] \tag{1}$$
$$= \min_{\pi \in \Pi(p,q)} \mathbb{E}_\pi[\|X - Y\|^2] ,$$

*The first definition is defined as the* Monge *formulation (Monge, 1781), and the second one as the* Kantorovitch *formulation (Kantorovich, 1942), with the equivalence holding when $p, q$ are absolutely continuous, or discrete uniform samples of the same finite size.*

Note that this distance can be approximated with sample access to $p$ and $q$ by plugging directly these samples empirical measures $\hat{p}_n$ of the $X_j$ and $\hat{q}_n$ of the $Y_j$. However, this approach suffers from two issues: It suffers from a curse of dimensionality, and the convergence of the estimate $W_2^2(\hat{p}_n, \hat{q}_n)$ to $W_2^2(p,q)$ is slow, in $n^{-1/d}$ (Dudley, 1969), it is slow to compute in general, with a worst case super cubic cost of $n^3$ for the Hungarian algorithm, and methods based on the Sinkhorn algorithm in $n^2 \log(n)$. The latter is motivated by an entropic-regularized formulation (Cuturi, 2013)

$$W_{2,\varepsilon}^2 = \min_{\pi \in \Pi(p,q)} \mathbb{E}_\pi[\|X - Y\|^2] - \varepsilon H(\pi) .$$

An interesting exception is the one-dimensional case: when $d = 1$, the solution of (1) is given by

$$W_2^2(\hat{p}_n, \hat{q}_n) = \frac{1}{n} \sum_{j=1}^n |\text{sort}(x)_j - \text{sort}(y)_j|^2$$

where sort : $\mathbb{R}^n \to \mathbb{R}^n$ is the function that maps a vector $x \in \mathbb{R}^n$ to its copy sorted in nondecreasing order $\text{sort}(x) = (x_{\sigma(1)}, \dots, x_{\sigma(n)})^\intercal$ such that $x_{\sigma(1)} \leq \dots \leq x_{\sigma(n)}$ (ties do not make this function ambiguous). It can therefore be computed in time of order $n \log n$. This can be leveraged for $d > 1$ by considering the average Wasserstein distance over uniformly random unit directions. This is called the *sliced Wasserstein distance*

**Definition A.3.** *The sliced Wasserstein distance (Rabin et al., 2011; Nadjahi, 2021) is defined as the average of the Wasserstein distances over 1-d projections along $u \sim \mathcal{U}(S)$ a uniformly random unit direction*

$$SW_2^2(p,q) = \mathbb{E}_{u \sim \mathcal{U}(S)}[W_2^2(u^\intercal p, u^\intercal q)]$$

*where $u^\intercal p$ (resp. $u^\intercal q$) denotes the distribution of $u^\intercal X$ when $X \sim p$ (resp. $u^\intercal Y$ when $Y \sim q$).*

The sliced Wasserstein distance is still a distance between distributions. It can also be easily estimated from samples, given empirical measures $\hat{p}_n$ and $\hat{q}_n$ and $M$ i.i.d. unit vectors $u_1, \dots u_M$

$$\hat{SW}_{2,M}^2(\hat{p}_n, \hat{q}_n) = \frac{1}{M} \sum_{i=1}^M W_2^2(u_i^\intercal \hat{p}_n, u_i^\intercal \hat{q}_n)$$
$$= \frac{1}{Mn} \sum_{i=1}^M \sum_{j=1}^n |\text{sort}(u_i^\intercal X)_j - \text{sort}(u_i^\intercal Y)_j|^2 ,$$

One of the advantages of this approach is the relaxed computational load: computing Wasserstein distances in 1D only requires to sort all the elements in the sample, which can be done in order of $n \log n$ time, and is highly parallelizable, allowing to perform $M$ of these operations with little to no overhead, and $M$ projections from dimension $d$ to 1, for $n$ points each time.

In particular, under mild assumptions, the sample complexity of estimating the sliced Wasserstein distance does not depend on the dimension of the problem (Nadjahi et al., 2020), in contrast to the standard Wasserstein distance for which the sample complexity grows exponentially with the dimension.

We finally note that as in Rabin et al. (2011), we consider the average of the *squared* 1-D Wasserstein distances, and would do so for other $\ell_p$, $p \neq 2$ norm costs.

### A.3 Metric comparison

**Remarks about FID**

- The last formula is also found in the literature as the following, both are equal

$$W_2^2(\mathcal{N}(\mu_X, \Sigma_X), \mathcal{N}(\mu_Y, \Sigma_Y)) = \|\mu_X - \mu_Y\|^2 + \mathrm{tr}(\Sigma_X + \Sigma_Y - 2(\Sigma_Y^{1/2}\Sigma_X\Sigma_Y^{1/2})^{1/2}) \,.$$

- In practice, the expectations are obtained based on a finite sample $X_1, \dots, X_n$ from a generative model and $Y_1, \dots, Y_n$ from a dataset, and we actually compute the plug-in estimate

$$\mathrm{FID}(\hat{p}_{n,\theta}, \hat{p}_{n,\mathrm{data}}) = \|\hat{\mu}_X - \hat{\mu}_Y\|^2 + \mathrm{tr}(\hat{\Sigma}_X + \hat{\Sigma}_Y - 2(\hat{\Sigma}_Y\hat{\Sigma}_X)^{1/2}) \,.$$

- This distance is motivated by the *Wasserstein distance* $W_2^2$ (see, e.g. Villani, 2008, and Appendix A.2 in this work), obtained by solving an optimal transport problem, with a square Euclidean distance cost. For FID, this distance is applied to two fitted Gaussian distributions rather than to the sample distributions $\hat{p}_{n,\theta}$ and $\hat{p}_{n,\mathrm{data}}$.

- Using this method, rather than a sample-based estimate of $W_2^2(\hat{p}_{n,\theta}, \hat{p}_{n,\mathrm{data}})$, allows to overcome the two main obstacles when using the Wasserstein distance between two distributions based on sample access: *statistical* and *computational complexity*. Computing the FID only requires to estimate the mean and covariance matrices, and to perform a conceptually simple, closed-form computation.

**Definition A.4** (mean FID)**.** *Let $X = \psi_w(g_\theta(Z)) \sim p_\theta$ and $Y = \psi_w(D) \sim p_{data}$,*

$$\mu_X = \mathbb{E}_{p_\theta}[X] \,, \quad \mu_Y = \mathbb{E}_{p_{data}}[Y] \,.$$

*The mean FID (that we denote by $\mu$FID) is defined as*

$$\mu FID(p_\theta, p_{data}) = \|\mu_X - \mu_Y\|^2 \,.$$

**Remarks**

- Much like for FID, it is also very easy to estimate the mean FID from a finite sample with the plug-in empirical measures $\mu\mathrm{FID}(\hat{p}_{n,\theta}, \hat{p}_{n,\mathrm{data}}) = \|\hat{\mu}_X - \hat{\mu}_Y\|^2$.

- We show in Section 4.4 that the sample complexity of $\mu$FID is much lower than that of FID - this probably stems from the fact that only a vector of size $d$ must be evaluated rather than a $d$-by-$d$ matrix.

- We show in Section 4.5 that it is even less robust than FID.

**Definition A.5** (Sliced FID)**.** *Let $X = \psi_w(g_\theta(Z)) \sim p_\theta$ and $Y = \psi_w(D) \sim p_{data}$, the sliced FID (that we denote by $\sigma$FID) is defined as*

$$\sigma FID(p_\theta, p_{data}) = \mathbb{E}_{u \sim \mathcal{U}(S)}[FID(u^\intercal p_\theta, u^\intercal p_{data})] \,.$$

*For finite samples* $(X_j)_{j\in[n]}$, $(Y_j)_{j\in[n]}$ *and* $(u_i)_{i\in[M]}$ *it can be estimated by*

$$\begin{aligned}\sigma\mathit{FID}(\hat{p}_{n,\theta}, \hat{p}_{n,\mathit{data}}) &= \frac{1}{M}\sum_{i=1}^{M} \mathit{FID}(u_i^\intercal \hat{p}_n, u_i^\intercal \hat{q}_n) \\ &= \frac{1}{M}\sum_{i=1}^{M} \left\{(u_i^\intercal \hat{\mu}_{n,X} - u_i^\intercal \hat{\mu}_{n,Y})^2 \right. \\ &\quad \left. + (\hat{\sigma}_{n,u_i^\intercal X} - \hat{\sigma}_{n,u_i^\intercal Y})^2\right\}.\end{aligned}$$

**Remarks**

- While very easy to estimate from samples, we also show that it suffers from the same robustness issues as the FID.

**Definition A.6** (Sinkhorn divergence (Genevay et al., 2018))**.** *For two distributions, we denote by* $W_\varepsilon(p, q)$ *the value of the entropic-regularized optimal transport problem between* $p$ *and* $q$ *(see, e.g. Peyré and Cuturi, 2019, and Appendix A.2 in this work). Let* $X = \psi_w(g_\theta(Z)) \sim p_\theta$ *and* $Y = \psi_w(D) \sim p_{\mathit{data}}$*, the Sinkhorn Divergence Inception Distance (SDID) is defined by*

$$\mathit{SDID}_\varepsilon(p_\theta, p_{\mathit{data}}) = W_\varepsilon(p_\theta, p_{\mathit{data}}) - \frac{1}{2}W_\varepsilon(p_\theta, p_\theta) - \frac{1}{2}W_\varepsilon(p_{\mathit{data}}, p_{\mathit{data}}).$$

**Remarks**

- For finite samples, the empirical measures $\hat{p}_{n,\theta}, \hat{p}_{n,\text{data}}$ can be split in $\hat{p}_{1,n,\theta}, \hat{p}_{2,n,\theta}, \hat{p}_{1,n,\text{data}}, \hat{p}_{2,n,\text{data}}$ and the divergence can be estimated by

$$\begin{aligned}\text{SDID}_\varepsilon(\hat{p}_{n,\theta}, \hat{p}_{n,\text{data}}) &= W_\varepsilon(\hat{p}_{1,n,\theta}, \hat{p}_{1,n,\text{data}}) \\ &\quad - \frac{1}{2}W_\varepsilon(\hat{p}_{1,n,\theta}, \hat{p}_{2,n,\theta}) \\ &\quad - \frac{1}{2}W_\varepsilon(\hat{p}_{1,n,\text{data}}, \hat{p}_{2,n,\text{data}}).\end{aligned}$$

  SDID can be computed by computing each entropic regularized optimal transport problem with a fast GPU-friendly alternate projection method, called Sinkhorn's algorithm (Cuturi, 2013).

- In practice, to overcome a curse of dimensionality, we have found it better to estimate the correction term from two independent samples $\hat{p}_{1,n}, \hat{p}_{2,n}$. This concretely doubles the required sample size. We have found this metric to be much more robust than FID, and to require a smaller sample size, but of a similar order (ignoring this doubling) - see Section 4.4.

**Definition A.7** (Maximum mean discrepancy - MMD (Gretton et al., 2012; Jayasumana et al., 2024))**.** *Let* $X = \psi_w(g_\theta(Z)) \sim p_\theta$ *and* $Y = \psi_w(D) \sim p_{\mathit{data}}$*, and the kernel function* $k_\sigma(x, y) = \exp(-\|x - y\|^2/\sigma)$*, for* $\sigma > 0$*, the MMD is defined as*

$$\begin{aligned}\mathit{MMD}(p_\theta, p_{\mathit{data}}) &= \mathbb{E}_{p_\theta \otimes p_\theta}[k(x, x')] - 2\mathbb{E}_{p_\theta \otimes p_{\mathit{data}}}[k(x, y)] \\ &\quad + \mathbb{E}_{p_{\mathit{data}} \otimes p_{\mathit{data}}}[k(y, y')].\end{aligned}$$

**Remarks**

- Since it is defined as a two-sample mean, the MMD can also be estimated quickly from empirical distributions $\hat{p}_{n,\theta}$ and $\hat{p}_{n,\text{data}}$.

- One of the drawbacks of this metric is the need to select a hyperparameter $\sigma > 0$.

- Another drawback is the computational aspect, as an $n \times n$ kernel matrix must be computed.

- The Sinkhorn divergence and MMD are related: when $\varepsilon \to +\infty$, we have that

$$\text{SDID}_\varepsilon \to \frac{1}{2}\text{MMD}_{-\|\cdot\|^2}$$

where the kernel function $k$ is given by the negative squared Euclidean distance (rather than a Gaussian kernel).

- The use of this metric, with another embedding network, is recommended in (Jayasumana et al., 2024).

# B Proofs

## B.1 Proof of Proposition 4.1

*Proof.* Recall that the target distribution $p$ has mean $\mu$ and covariance $\Sigma$. We construct the discrete distribution $\hat{q}$ using $2r$ vectors $v_i^{(+)} = \mu + \alpha u_i$ and $v_i^{(-)} = \mu - \alpha u_i$, where each vector is assigned probability $\pi_i = \lambda_i/(2\,\mathrm{tr}(\Sigma))$, and $\alpha = \sqrt{\mathrm{tr}(\Sigma)}$.

**1. Mean of $\hat{q}$** By the definition of expectation for a discrete distribution:

$$\begin{aligned}\mathbb{E}_{\hat{q}}[v] &= \sum_{i=1}^{r} \pi_i v_i^{(+)} + \sum_{i=1}^{r} \pi_i v_i^{(-)} \\ &= \sum_{i=1}^{r} \pi_i(\mu + \alpha u_i) + \sum_{i=1}^{r} \pi_i(\mu - \alpha u_i) \\ &= \sum_{i=1}^{r} \pi_i(2\mu) = \mu \sum_{i=1}^{r} 2\pi_i.\end{aligned}$$

Since $2\pi_i = \lambda_i/\,\mathrm{tr}(\Sigma)$ and $\sum_{i=1}^{r} \lambda_i = \mathrm{tr}(\Sigma)$, it follows that $\sum 2\pi_i = 1$, hence $\mathbb{E}_{\hat{q}}[v] = \mu$.

**2. Covariance of $\hat{q}$** The covariance of $\hat{q}$ is given by $\mathbb{E}_{\hat{q}}[(v-\mu)(v-\mu)^\intercal]$:

$$\begin{aligned}\mathrm{Cov}_{\hat{q}}(v) &= \sum_{i=1}^{r} \pi_i (v_i^{(+)} - \mu)(v_i^{(+)} - \mu)^\intercal + \sum_{i=1}^{r} \pi_i (v_i^{(-)} - \mu)(v_i^{(-)} - \mu)^\intercal \\ &= \sum_{i=1}^{r} \pi_i (\alpha u_i)(\alpha u_i)^\intercal + \sum_{i=1}^{r} \pi_i (-\alpha u_i)(-\alpha u_i)^\intercal \\ &= \sum_{i=1}^{r} 2\pi_i \alpha^2 u_i u_i^\intercal.\end{aligned}$$

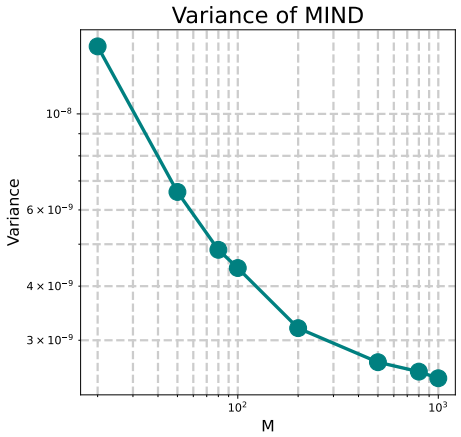
Substituting $\alpha^2 = \mathrm{tr}(\Sigma)$ and $2\pi_i = \lambda_i/\,\mathrm{tr}(\Sigma)$:

$$\begin{aligned}\mathrm{Cov}_{\hat{q}}(v) &= \sum_{i=1}^{r} \left(\frac{\lambda_i}{\mathrm{tr}(\Sigma)}\right) \mathrm{tr}(\Sigma) u_i u_i^\intercal \\ &= \sum_{i=1}^{r} \lambda_i u_i u_i^\intercal = \Sigma.\end{aligned}$$

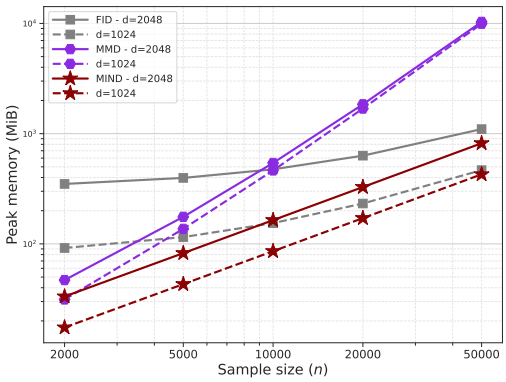
**3. FID Value** The FID between two distributions is defined as the 2-Wasserstein distance between their associated Gaussians (Heusel et al., 2017). Consequently, FID is strictly a function of the first two moments. Since $\mathbb{E}_{\hat{q}}[v] = \mu_p$ and $\mathrm{Cov}_{\hat{q}}(v) = \Sigma_p$, the means and covariances match exactly:

$$\begin{aligned}\mathrm{FID}(\hat{q}, p) &= \|\mu - \mu\|^2 + \mathrm{tr}(\Sigma + \Sigma - 2(\Sigma\Sigma)^{1/2}) \\ &= 0 + \mathrm{tr}(2\Sigma - 2\Sigma) = 0.\end{aligned}$$

This concludes the proof. □




(a) Variance of the MIND with different number of projections $M$.




(b) Peak memory used for calculating different metrics.

# C Additional results

## C.1 Effects of number of projections

As shown in Figure 10a, our empirical analysis shows that the variance is not affected at smaller scales than numerical artifacts for $M > 1000$. We also show that using $M = 100$ yields almost the same performance, while there is a substantial degradation when using $M = 10$.

## C.2 Peak Memory

The measurements include both memory occupied by the input data and the temporary memory required for metric computation. As shown in Figure 10b, MIND at its recommended sample size ($n = 5$k) requires less memory than MMD and FID.

## C.3 Computation resources and experimental details

We run each experiment in Section 4.4 for 2 hours using 4 TPUv5e and each experiment in Section 4.5 for 10 minutes using 4 TPUv5e. The diffusion model used in Section 4.4 is trained with 5M steps on ImageNet-64, we summarize other details for its training and sampling in Table 2.

| Name | Value |
|---|---|
| Condition embedding dimension | 512 |
| Noise embedding dimension | 512 |
| Optimizer | Adam with standard hyperparameters |
| Learning Rate | $10^{-5}$ |
| EMA decay | 0.9999 |
| Hardware | 16 TPUv6e |
| Noise schedule | Rectified Flow |
| Number of sampling steps | 250 |
| CFG weight | 0.0 |
| Number of base channels | 192 |
| Attention Head Dimension | 64 |
| Number of downsampling | 4 |
| Channels multiplier | (1, 2, 3, 4) |
| Residual blocks per level | (3, 3, 3, 3) |

Table 2: Hyperparameters for training and sampling from diffusion models.

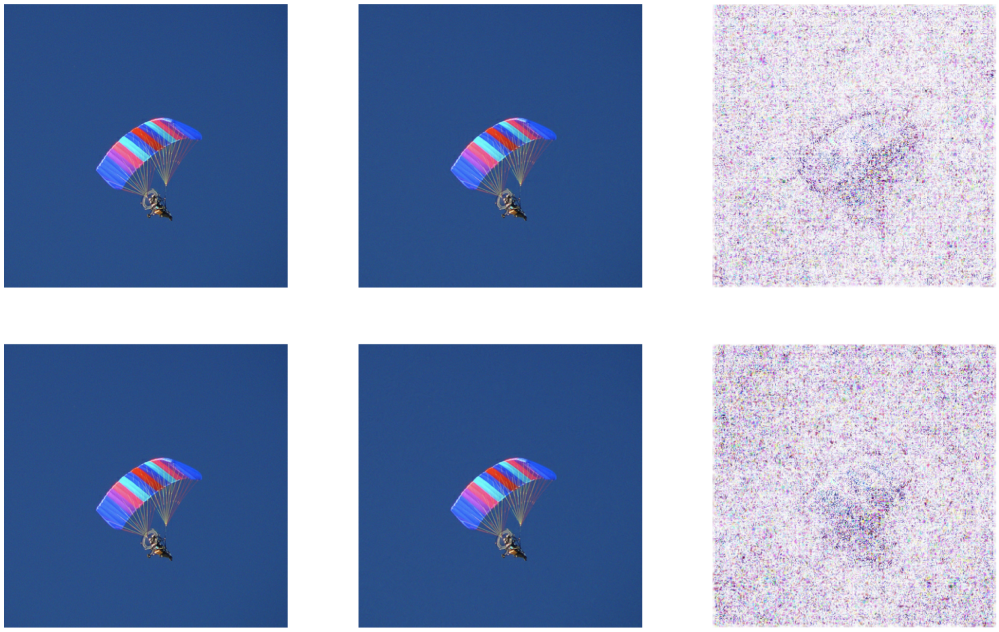


Figure 11: Two elements of the batch, all initial images are the same. (**Left**) Initial image (**Center**) Image after optimization (**Right**) Difference scaled by a factor 100 (to become visible).

### C.4 Moment-matching hacking

We illustrate the results of the experiment described in Section 4.5 in Figure 11