MIND: Monge Inception Distance for Generative Models Evaluation

Q. Berthet, C. Crepy, R. Elie, K. Greff, M. Sander, Y.-H. Wu

Google DeepMind

TL;DR

We propose the Monge Inception Distance (`MIND`), a metric for evaluating generative model that addresses key limitations of the widely adopted Fréchet Inception Distance (FID).

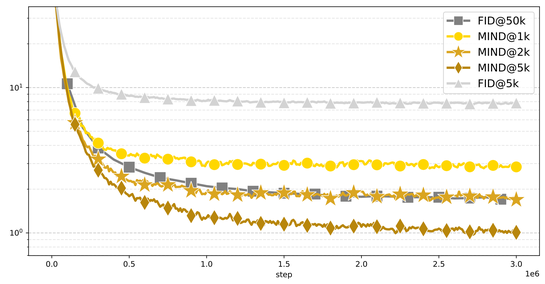
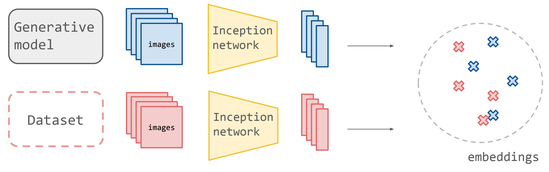


Figure: `MIND` metric during a diffusion model trainng on on ImageNet64 (log scale), illustrating how $\texttt{MIND}_{2k}$ can be used to replace FID_{50k}.

Evaluation of generative model



1. Generate some samples (typically 50k) with the target generative model.
2. Obtain features for generated samples and true samples via feature extraction model (e.g. Inception-v3 model (Salimans et al., 2016))
3. Calculate the distributional distance of generated features vs. true features.

Fréchet Inception Distance (FID)

FID (Heusel et al., 2017) measures the distributional distance based on their first two moments: Denote μ_X, Σ_X and μ_Y, Σ_Y be the means and covariances of p_θ and p_{data}. Then

$$\text{FID}(p_\theta, p_{\text{data}}) = \|\mu_X - \mu_Y\|_2^2 + \text{tr}(\Sigma_X + \Sigma_Y - 2(\Sigma_Y\Sigma_X)^{1/2}) .$$

Drawbacks of FID

- Computation of FID is rank deficient for $n \leqslant d$. Therefore, for images, the sample size usually used is 10k or 50k.
- FID is not a proper distance. Two distributions can share identical mean and covariance matrix while being very different.

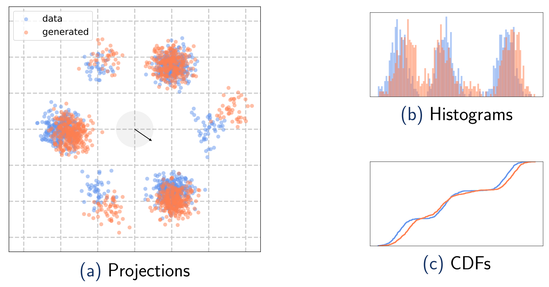
Monge Inception Distance - `MIND`

Inspired by sliced Wasserstein distance (Rabin et al., 2011), the Monge inception distance `MIND` is given by averaging W_2^2 square distances for projections of the distributions along unit directions. For finite samples $(X_j)_{j\in[n]}$, $(Y_j)_{j\in[n]}$, random unit directions $(u_i)_{i\in[M]}$ and $\alpha > 0$, it is given by

$$\begin{aligned}\texttt{MIND}(\hat{p}_{n,\theta}, \hat{p}_{n,\text{data}}) &= \frac{\alpha}{M}\sum_{i=1}^{M} W_2^2(u_i^\top \hat{p}_n, u_i^\top \hat{q}_n) , \\ &= \frac{\alpha}{nM}\sum_{i=1}^{M}\sum_{j=1}^{n} |\text{sort}(u_i^\top X)_j - \text{sort}(u_i^\top Y)_j|^2 .\end{aligned}$$

Advantage of `MIND`

- Requires less samples (5k instead of 50k).
- Faster computation.
- More robust.



(a) Projections

(b) Histograms

(c) CDFs

Figure: Sliced Wasserstein in `MIND`, one projection. (**Left**) Two samples of synthetic embeddings (red and blue), along with the unit sphere and a random unit direction u. (**Middle**) The two histograms of the projections along u. (**Right**) The associated cumulative distribution functions (cdf).

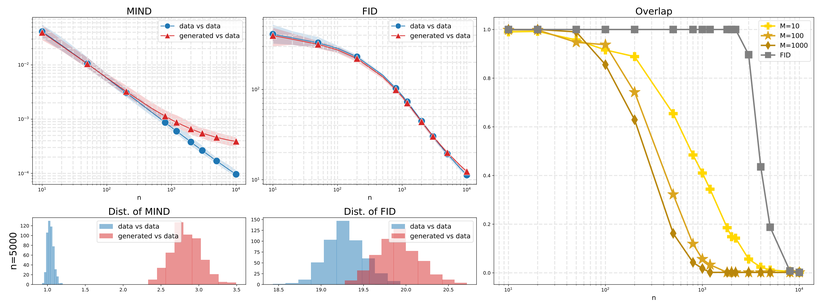
```python
def monge_inception_distance(x, y, theta):
  """MIND metric.

  Args:
    x: Input generated features.
    y: Ground truth features.
    theta: Projection matrix.

  Returns:
    The value of the MIND metric.

  """
  num_samples = x.shape[0]

  x_proj = theta @ x.T
  y_proj = theta @ 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 jnp.mean(dists)
```

Figure: JAX implementation of `MIND`, leveraging parralelized sorting with `jax.lax.top_k`.

Generated vs. true data

For various sample sizes n, we evaluate for all metrics Δ (e.g. FID, `MIND`),

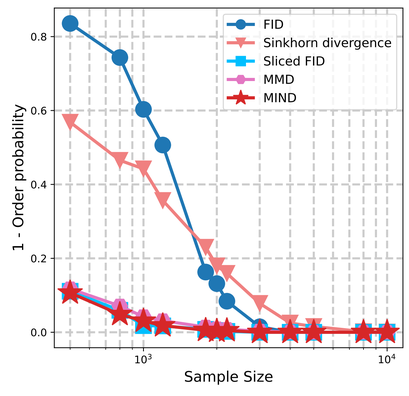
$$\mathbb{P}\big(\Delta(\hat{p}_{n,\text{data}}, \hat{p}'_{n,\text{data}}) \geqslant \Delta(\hat{p}_{n,\text{data}}, \hat{p}_{n,\text{generated}})\big) .$$



Monotonicity in training

For various sample sizes n, we evaulate for all metrics Δ at k different checkpoints $\theta_1, \dots, \theta_k$, where they are ordered by the time of training (i.e. θ_1 the earliest and θ_k the latest),

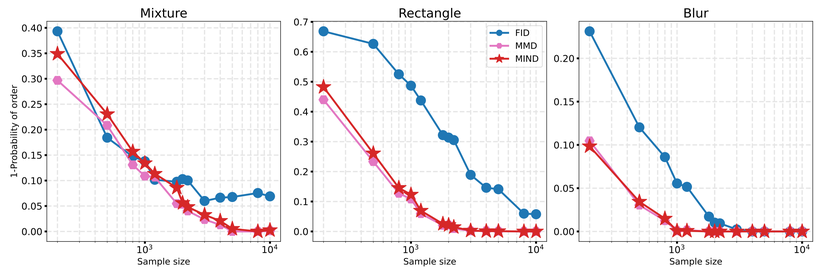
$$1 - \mathbb{P}\big(\Delta(\hat{p}_{n,\text{data}}, \hat{p}_{n,\theta_1}) \geqslant \dots \geqslant \Delta(\hat{p}_{n,\text{data}}, \hat{p}_{n,\theta_k})\big) .$$



Clean vs. perturbed generated samples

For various sample sizes n, we evaluate for all metrics Δ at k levels of perturbation $\varepsilon_1, \dots, \varepsilon_k$,

$$1 - \mathbb{P}\big(\Delta(\hat{p}_{n,\text{data}}, \hat{p}_{n,\text{data},\varepsilon_1}) \geqslant \dots \geqslant \Delta(\hat{p}_{n,\text{data}}, \hat{p}_{n,\text{data},\varepsilon_k})\big) .$$



Moment Matching

For $n = 2r$ and a batch $a^0 \in \mathbb{R}^{2r\times[\text{dims}]}$ of $2r$ images, each of shape [dims] (e.g. $[512, 512, 3]$), and a target distribution p_{data} over $\mathbb{R}^{[\text{dims}]}$, we found a way to match the first and second moments (μ_{a^0} and Σ_{a^0}) of the batch to the first and second moments of p_{data}.

Robustness of metrics

Metric	ratio
FID	11.2%
μFID	2.6%
σFID	4.2%
MMD	12.2%
`MIND`	31.1%

Table: Robustness of several metrics under moment matching

Computation time comparison

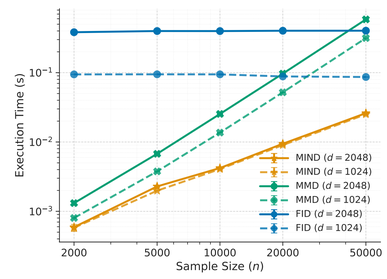


Figure: Walltime comparison for `MIND`, MMD, and FID

References

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017).Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems, 30.*

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

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016).Improved techniques for training GANs. *Advances in Neural Information Processing Systems (NIPS).*