A Organization of the supplementary material

In Section B, we describe in details the training and sampling procedures for DMMD. In Section C, we describe more details for the 2d experiments. In Section E, we provide more details about DKALE-Flow method. In Section F, we provide experimental details for the image datasets. In Section H, we provide proof for the theoretical results described in Section 3 from the main section of the paper. Finally, in Section I we present the samples from DMMD on different image datasets.

B DMMD training and sampling

B.1 MMD discriminator

Let $\mathcal{X} \subset \mathbb{R}^D$ and $\mathcal{P}(\mathcal{X})$ be the set of probability distributions defined on $\mathcal{X}$. Let $P \in \mathcal{P}(\mathcal{X})$ be the *target* or data distribution and $Q_\psi \in \mathcal{P}(\mathcal{X})$ be a distribution associated with a *generator* parameterized by $\psi \in \mathbb{R}^L$. Let $\mathcal{H}$ be Reproducing Kernel Hilbert Space (RKHS), see (Schölkopf & Smola, 2018) for details, for some kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) between Q_ψ and P is defined as $\mathrm{MMD}(Q_\psi, P) = \sup_{f\in\mathcal{H}}\{\mathbb{E}_{Q_\psi}[f(X)] - \mathbb{E}_P[f(X)]\}$. Given $X^N = \{x_i\}_{i=1}^N \sim Q_\psi^{\otimes N}$ and $Y^M = \{y_i\}_{i=1}^M \sim P^{\otimes M}$, an unbiased estimate of MMD^2 (Gretton et al., 2012) is given by

$$\mathrm{MMD}_u^2[X^N, Y^M] = \frac{1}{N(N-1)}\sum_{i\neq j}^N k(x_i, x_j) + \frac{1}{M(M-1)}\sum_{i\neq j}^M k(y_i, y_j) - \frac{2}{NM}\sum_{i=1}^N\sum_{j=1}^M k(x_i, y_j). \tag{14}$$

In MMD GAN (Bińkowski et al., 2021; Li et al., 2017), the kernel in the objective (14) is given as

$$k(x,y) = k_{\text{base}}(\phi(x;\theta), \phi(y;\theta)), \tag{15}$$

where k_{base} is a base kernel and $\phi(\cdot;\theta) : \mathcal{X} \to \mathbb{R}^K$ are neural networks *discriminator* features with parameters $\theta \in \mathbb{R}^H$. We use the modified notation of $\mathrm{MMD}_u^2[X^N, Y^M; \theta]$ for equation (14) to highlight the functional dependence on the discriminator parameters. MMD is an instance of Integral Probability Metric (IPM) (see (Arjovsky et al., 2017)) which is well defined on distributions with disjoint support unlike f-divergences (Nowozin et al., 2016). An advantage of using MMD over other IPMs (see for example, Wasserstein GAN (Arjovsky et al., 2017)) is the flexibility to choose a kernel k. Another form of MMD is expressed as a norm of a *witness function*

$$\mathrm{MMD}(Q_\psi, P) = \sup_{f\in\mathcal{H}}\{\mathbb{E}_{Q_\psi}[f(X)] - \mathbb{E}_P[f(X)]\} = \|f_{Q_\psi,P}\|_{\mathcal{H}},$$

where the witness function $f_{Q_\psi,P}$ is given as

$$f_{Q_\psi,P}(z) = \int k(x,z)dQ_\psi - \int k(y,z)dP(y)$$

Given two sets of samples $X^N = \{x_i\}_{i=1}^N \sim Q_\psi^{\otimes N}$ and $Y^M = \{y_i\}_{i=1}^M \sim P^{\otimes M}$, and the kernel (15), the empirical witness function is given as

$$\hat{f}_{Q_\psi,P}(z) = \frac{1}{N}\sum_{i=1}^N k_{\text{base}}(\phi(x_i;\theta), \phi(z;\theta)) - \frac{1}{M}\sum_{j=1}^M k_{\text{base}}(\phi(y_j;\theta), \phi(z;\theta))$$

The ℓ_2 penalty (Bińkowski et al., 2021) is defined as

$$\mathcal{L}_{\ell_2}(\theta) = \frac{1}{N}\sum_{i=1}^N \|\phi(x_i;\theta)\|_2^2 + \frac{1}{N}\sum_{i=1}^N \|\phi(y_i;\theta)\|_2^2$$

Assuming that $M = N$ and following (Bińkowski et al., 2021; Gulrajani et al., 2017), for $\alpha_i \sim U[0,1]$, where $U[0,1]$ is a uniform distribution on $[0,1]$, we construct $z_i = x_i\alpha_i + (1-\alpha)y_i$ for all $i = 1, \ldots, N$. Then, the gradient penalty (Bińkowski et al., 2021; Gulrajani et al., 2017) is defined as

$$\mathcal{L}_\nabla(\theta) = \frac{1}{N}\sum_{i=1}^N (\|\nabla \hat{f}_{Q_\psi,P}(z_i)\|_2 - 1)^2$$

We denote by $\mathcal{L}(\theta)$ the MMD discriminator loss given as

$$\mathcal{L}(\theta) = -\mathrm{MMD}_u^2[X^N, Y^M; \theta] = \tfrac{1}{N(N-1)} \sum_{i \neq j}^{N} k_{\text{base}}(\phi(x_i; \theta), \phi(x_j; \theta)) + \tfrac{1}{M(M-1)} \sum_{i \neq j}^{M} k_{\text{base}}(\phi(y_i; \theta), \phi(y_j; \theta)) \\ - \tfrac{2}{NM} \textstyle\sum_{i=1}^{N} \sum_{j=1}^{M} k_{\text{base}}(\phi(x_i; \theta), \phi(y_j; \theta))$$

Then, the total loss for the discriminator on the two samples of data assuming that $N = M$ is given as

$$\mathcal{L}_{\text{tot}}(\theta) = \mathcal{L}(\theta) + \lambda_{\nabla} \mathcal{L}_{\nabla}(\theta) + \lambda_{\ell_2} \mathcal{L}_{\ell_2}(\theta),$$

for some constants $\lambda_{\nabla} \geq 0$ and $\lambda_{\ell_2} \geq 0$.

B.2 Noise-dependent MMD

In Section 4, we describe the approach to train MMD discriminator from forward diffusion using noise-dependent discriminators. For that, we assume that we are given a noise level $t \sim U[0, 1]$ where $U[0, 1]$ is a uniform distribution on $[0, 1]$, and a set of clean data $X^N = \{x^i\}_{i=1}^N \sim P^{\otimes N}$. Then we produce a set of noisy samples x_t^i using forward diffusion process (6). We denote these samples by $X_t^N = \{x_t^i\}_{i=1}^N$. We define noise conditional kernel

$$k(x, y; t, \theta) = k_{\text{base}}(\phi(x, t; \theta), \phi(y, t; \theta)),$$

with noise conditional features $\phi(x, t; \theta)$. This allows us to define the noise conditional discriminator loss

$$\mathcal{L}(\theta, t) = -\mathrm{MMD}_u^2[X^N, X_t^N, t, \theta] = \tfrac{1}{N(N-1)} \sum_{i \neq j}^{N} k_{\text{base}}(\phi(x_t^i; t, \theta), \phi(x_t^j; t, \theta)) + \\ \tfrac{1}{N(N-1)} \sum_{i \neq j}^{N} k_{\text{base}}(\phi(x^i; t, \theta), \phi(x^j; t, \theta)) \\ - \tfrac{2}{N^2} \textstyle\sum_{i=1}^{N} \sum_{j=1}^{N} k_{\text{base}}(\phi(x^i; t, \theta), \phi(x_t^j; t, \theta)) \tag{16}$$

The noise conditional ℓ_2 penalty is given as

$$\mathcal{L}_{\ell_2}(\theta, t) = \frac{1}{N} \sum_{i=1}^{N} \|\phi(x_t^i; t, \theta)\|_2^2 + \frac{1}{N} \sum_{i=1}^{N} \|\phi(x^i; t, \theta)\|_2^2$$

The noise conditional gradient penalty is given as

$$\mathcal{L}_{\nabla}(\theta, t) = \frac{1}{N} \sum_{i=1}^{N} (\|\nabla \hat{f}_{P,t}(z_i)\|_2 - 1)^2,$$

where $z_i = \alpha_i x_t^i + (1 - \alpha_i) x^i$ for $\alpha_i \sim U[0, 1]$ and the noise conditional witness function

$$\hat{f}_{P,t}(z) = \frac{1}{N} \sum_{i=1}^{N} k_{\text{base}}(\phi(x_i^t; t, \theta), \phi(z; \theta)) - \frac{1}{N} \sum_{j=1}^{N} k_{\text{base}}(\phi(x_i; t, \theta), \phi(z; \theta)) \tag{17}$$

Therefore, the total noise conditional loss is given as

$$\mathcal{L}_{\text{tot}}(\theta, t) = \mathcal{L}(\theta, t) + \lambda_{\nabla} \mathcal{L}_{\nabla}(\theta, t) + \lambda_{\ell_2} \mathcal{L}_{\ell_2}(\theta, t), \tag{18}$$

for some constants $\lambda_{\nabla} \geq 0$ and $\lambda_{\ell_2} \geq 0$.

B.3 Linear kernel for scalable MMD

Computational complexity of (18) is $O(N^2)$. Here, we assume that the base kernel is linear, i.e.

$$k_{\text{base}}(x, y) = \langle x, y \rangle$$

This allows us to simplify the MMD computation (16) as

$$
\begin{aligned}
\mathrm{MMD}_u^2[X^N, X_t^N, t, \theta] = \frac{1}{N(N-1)} \left(\bar{\phi}_t(X_t^N)^T \bar{\phi}_t(X_t^N) - \|\bar{\phi_t}\|^{2^N}(X_t) \right) + \\
\frac{1}{N(N-1)} \left(\bar{\phi}_t(X^N)^T \bar{\phi}_t(X^N) - \|\bar{\phi_t}\|^{2^N}(Y) \right) \\
- \frac{2}{NN} (\bar{\phi}_t(X_t^N))^T \bar{\phi}_t(X^N),
\end{aligned}
\tag{19}
$$

where

$$
\begin{aligned}
\bar{\phi}_t(X_t^N) &= \sum_{i=1}^N \phi(x_t^i; \theta_t) \\
\bar{\phi}_t(X^N) &= \sum_{j=1}^N \phi(x^i; \theta_t) \\
\|\bar{\phi_t}\|^2(X_t^N) &= \sum_{i=1}^N \|\phi(x_t^i; \theta_t)\|^2 \\
\|\bar{\phi_t}\|^2(X^N) &= \sum_{j=1}^N \|\phi(x^i; \theta_t)\|^2
\end{aligned}
$$

Therefore we can pre-compute quantities $\bar{\phi}_t(X_t^N), \bar{\phi}_t(X^N), \|\bar{\phi_t}\|^2(X_t^N), \|\bar{\phi_t}\|^2(X^N)$ which takes $O(N)$ and compute $\mathrm{MMD}_u^2[X^N, X_t^N, t, \theta]$ in $O(1)$ time. This also leads $O(1)$ computation complexity for $\mathcal{L}_{\ell_2}$ and $O(N)$ complexity for $\mathcal{L}_{\nabla}$. This means that we simplify the computational complexity to $O(N)$ from $O(N^2)$.

At sampling, following (9) requires to compute the witness function (17) for each particle, which for a general kernel takes $O(N^2)$ in total. Using the linear kernel above, simplifies the complexity of the witness as follows

$$\hat{f}_{P,t}(z) = \langle \bar{\phi}_t(Z^N) - \bar{\phi}_t(X^N), \phi(z; \theta) \rangle,$$

where Z^N is a set of N noisy particles. We can precompute $\bar{\phi}_t(Z^N)$ in $O(N)$ time. Therefore one iteration of a witness function will take $O(1)$ time and for N noisy particles it makes $O(N)$.

B.4 Approximate sampling procedure

In this section we provide an algorithm for the approximate sampling procedure. The only change with the original Algorithm 2 is the approximate witness function

$$\hat{f}^\star_{P_t,P}(z) = \langle \phi(z, t; \theta^\star), \bar{\phi}(X_t, t, \theta^\star) - \bar{\phi}(X_0, t, \theta^\star) \rangle,$$

where

$$
\begin{aligned}
\bar{\phi}(X_0, t, \theta^\star &= \frac{1}{N} \sum_{i=1}^N \phi(x_0^i, t; \theta^\star) \\
\bar{\phi}(X_t, t, \theta^\star &= \frac{1}{N} \sum_{i=1}^N \phi(x_t^i, t; \theta^\star)
\end{aligned}
\tag{20}
$$

Here $x_0^i, i = 1, \dots, N$ correspond to the whole training set of clean samples and $x_t^i, i = 1, \dots,$ correspond to the noisy version of these clean samples produced by the forward diffusion process 6 for a given noise level t. These features can be precomputed once for every noise level t. The resulting algorithm is given in Algorithm (3). Another crucial difference with the original algorithm is the ability to run it for each particle Z independently.

Algorithm 3 Approximate noise-adaptive MMD gradient flow for a single particle

Inputs: T is the number of noise levels
$t_{\max}, t_{\min}$ are maximum and minimum noise levels
N_s is the number of gradient flow steps per noise level
$\eta > 0$ is the gradient flow learning rate
$\bar{\phi}(X_0, t, \theta^\star)$ - precomputed clean features for all $t = 1, \dots, T$ with (20)
$\bar{\phi}(X_t, t, \theta^\star)$ - precomputed noisy features for all $t = 1, \dots, T$ with (20)
Steps: Sample initial noisy particle $Z \sim \mathrm{N}(0, \mathrm{Id})$
for $i = T$ **to** 0 **do**
  Set the noise level $t = i\Delta t$ and $Z_0^t = Z$
  for $n = 0$ **to** $N_s - 1$ **do**
    $Z_{n+1}^t = Z_n^t - \eta \langle \nabla_z \phi(Z_n^t, t; \theta^\star), \bar{\phi}(X_t, t, \theta^\star) - \bar{\phi}(X_0, t, \theta^\star) \rangle$
  end for
  Set $Z = Z_N^t$
end for
Output Z

C Toy 2-D datasets experiments

For the 2-D experiments, we train DMMD using Algorithm (1) for $N_{\text{iter}} = 50000$ steps with a batch size of $B = 256$ and a number of noise levels per batch equal to $N_{\text{noise}} = 128$. The Gradient penalty constant $\lambda_\nabla = 0.1$ whereas the ℓ_2 penalty is not used. To learn noise-conditional MMD for DMMD, we use a 4-layers MLP $g(t; \theta)$ with ReLU activation to encode $\sigma(t; \theta) = \sigma_{\min} + \mathrm{ReLU}(g(t; \theta))$ with $\sigma_{\min} = 0.001$, which ensures $\sigma(t; \theta) > 0$. The MLP layers have the architecture of $[64, 32, 16, 1]$. Before passing the noise level $t \in [0, 1]$ to the MLP, we use sinusoidal embedding similar to the one used in (Ho et al., 2020), which produces a feature vector of size 1024. The forward diffusion process from (Ho et al., 2020) have modified parameters such that corresponding $\beta_1 = 10^-4, \beta_T = 0.0002$. On top of that, we discretize the corresponding process using only 1000 possible noise levels, with $t_{\min} = 0.05$ and $t_{\max} = 1.0$. At sampling time for the algorithm 2, we use $t_{\min} = 0.05, t_{\max} = 1.0$, $N_s = 10$ and $T = 100$. The learning rate $\eta = 1.0$. As basleines, we consider MMD-GAN with a generator parameterised by a 3-layer MLP with ELU activations. The architecture of the MLP is $[256, 256, 2]$. The initial noise for the generator is produced from a uniform distribution $U[-1, 1]$ with a dimensionality of 128. The gradient penalty coefficient equals to 0.1. As for the discriminator, the only learnable parameter is σ. We train MMD-GAN for 250000 iterations with a batch size of $B = 256$. Other variants of MMD gradient flow use the same sampling parameters as DMMD.

We used 1 $a100$ GPU with $40GB$ of memory to run these experiments. In total, all the experiments took less than 2 hours.

D f-divergences

The approach described in Section 4 can be applied to any divergence which has a well defined Wasserstein Gradient Flow described by a gradient of the associated witness function. Such divergences include the variational lower bounds on f-divergences, as described by (Nowozin et al., 2016), which are popular in GAN training, and were indeed the basis of the original GAN discriminator (Goodfellow et al., 2014). One such f-divergence is the KL Approximate Lower bound Estimator (KALE, Glaser et al., 2021). Unlike the original KL divergence, which requires a density ratio, the KALE remains well defined for distributions with non-overlapping support. Similarly to MMD, the Wasserstein Gradient of KALE is given by the gradient of a learned witness function. Thus, we train noise-conditional KALE discriminator and use corresponding noise-conditional Wasserstein gradient flow, as with DMMD. We call this method *Diffusion* KALE *flow* (D-KALE-Flow). This approach is described in Appendix E. We found this approach to lead to reasonable empirical results, but unlike with DMMD, it achieved worse performance than a corresponding GAN, see Appendix G.1.

E D-KALE-FLOW

In this section, we describe the DKALE-flow algorithm mentioned in Section D. Let $\mathcal{X} \subset \mathbb{R}^D$ and $\mathcal{P}(\mathcal{X})$ be the set of probability distributions defined on $\mathcal{X}$. Let $P \in \mathcal{P}(\mathcal{X})$ be the *target* or data distribution and $Q \in \mathcal{P}(\mathcal{X})$ be some distribution. The KALE objective (see (Glaser et al., 2021)) is defined as

$$KALE(Q, P|\lambda) = (1+\lambda)\max_{h\in\mathcal{H}}\{1 + \int hdQ - \int e^h dP - \frac{\lambda}{2}||h||^2_{\mathcal{H}}\}, \tag{21}$$

where $\lambda \geq 0$ is a positive constant and $\mathcal{H}$ is the RKHS with a kernel k. In practice, KALE divergence (21) can be replaced by a corresponding parametric objective

$$KALE(Q, P|\lambda, \theta, \alpha) = (1+\lambda)\left(\int h(X;\theta,\alpha)dQ(X) - \int e^{h(Y;\theta,\alpha)}dP(Y) - \frac{\lambda}{2}||\alpha||^2_2\right), \tag{22}$$

where

$$h(X;\theta,\alpha) = \phi(X;\theta)^T\alpha,$$

with $\phi(X;\theta) \in \mathbb{R}^D$ and $\alpha \in \mathbb{R}^D$. The objective function (22) can then be maximized with respect to θ and α for given Q and P. Similar to DMMD, we consider a noise-conditional witness function

$$h(x;t,\theta,\alpha,\psi) = \phi(x;t,\theta)^T\alpha(t;\psi)$$

From here, the noise-conditional KALE objective is given as

$$\mathcal{L}(\theta,\psi,t|\lambda) = KALE(P_t, P|\lambda,\theta,\alpha),$$

where P_t is the distribution corresponding to a forward diffusion process, see Section 4. Then, the total noise-conditional objective is given as

$$\mathcal{L}_{\text{tot}}(\theta,\psi,t|\lambda) = \mathcal{L}(\theta,\psi,t|\lambda) + \lambda_\nabla\mathcal{L}_\nabla(\theta,\psi,t) + \lambda_{\ell_2}\mathcal{L}_{\ell_2}(\theta,t),$$

where gradient penalty has similar form to WGAN-GP (Gulrajani et al., 2017)

$$\mathcal{L}_\nabla(\theta,\psi,t) = \mathbb{E}_Z(||\nabla_Z h(Z;t,\theta,\alpha,\psi)||_2 - 1)^2,$$

where $Z = \beta X + (1-\beta)Y$, $\beta \sim U[0,1]$, $X \sim P(X)$ and $Y \sim P(Y)$. The l2 penalty is given as

$$\mathcal{L}_{\ell_2}(\theta,t) = \frac{1}{2}\left(\mathbb{E}_{X\sim P(X)}||\phi(X;t,\theta)||^2 + \mathbb{E}_{Y\sim P(Y)}||\phi(Y;t,\theta)||^2\right)$$

Therefore, the final objective function to train the discriminator is

$$\mathcal{L}_{\text{tot}}(\theta,\psi|\lambda) = \mathbb{E}_{t\sim U[0,1]}\left[\mathcal{L}_{\text{tot}}(\theta,\psi,t|\lambda)\right]$$

This objective function depends on RKHS regularization λ, on gradient penalty regularization λ_∇ and on l2-penalty regularization λ_{ℓ_2}. Unlike in DMMD, we do not use an explicit form for the witness function and do not use the RKHS parameterisation. On one hand, this allows us to have a more scalable approach, since we can compute KALE and the witness function for each individual particle. On the other hand, the explicit form of the witness function in DMMD introduces beneficial inductive bias. In DMMD, when we train the discriminator, we only learn the kernel features, i.e. corresponding RKHS. In D-KALE, we need to learn both, the kernel features $\phi(x;t,\theta)$ as well as the RKHS projections $\alpha(t;\psi)$. This makes the learning problem for D-KALE more complex. The corresponding noise adaptive gradient flow for KALE divergence is described in Algorithm 4. An advantage over original DMMD gradient flow is the ability to run this flow individually for each particle.

F IMAGE GENERATION EXPERIMENTS

For the image experiments, we use CIFAR10 (Krizhevsky et al., 2009) dataset. We use the same forward diffusion process as in (Ho et al., 2020). As a Neural Network backbone, we use U-Net (Ronneberger et al., 2015) with a slightly modified architecture from (Ho et al., 2020). Our neural network architecture follows the backbone used in (Ho et al., 2020). On top of that we output the intermediate features at four levels – before down sampling, after down-sampling, before upsampling and a final layer. Each of these feature vectors is processed by a group normalization, the

Algorithm 4 Noise-adaptive KALE flow for single particle

Inputs: T is the number of noise levels
$t_{\text{max}}, t_{\text{min}}$ are maximum and minimum noise levels
N_{s} is the number of gradient flow steps per noise level
$\eta > 0$ is the gradient flow learning rate
Trained witness function $h(\cdot; t, \theta^\star, \psi^\star)$
Steps: Sample initial noisy particle $Z \sim \mathrm{N}(0, \mathrm{Id})$
Set $\Delta t = (t_{\text{max}} - t_{\text{min}})/T$
for $i = T$ **to** 0 **do**
  Set the noise level $t = t_{\text{min}} + i\Delta t$ and $Z_0^t = Z$
  for $n = 0$ **to** $N_{\text{s}} - 1$ **do**
    $Z_{n+1}^t = Z_n^t - \eta \nabla h(Z_n^t; t, \theta^\star, \psi^\star)$
  end for
  Set $Z = Z_N^t$
end for
Output Z

activation function and a linear layer producing an output vector of size 32. To produce the output of a discriminator features, these four feature vectors are concatenated to produce a final output feature vector of size 128. The noise level time is processed via sinusoidal time embedding similar to (Ho et al., 2020). We use a dropout of 0.2. DMMD is trained for $N_{\text{iter}} = 250000$ iterations with a batch size $B = 64$ with number $N_{\text{noise}} = 16$ of noise levels per batch. We use a gradient penalty $\lambda_\nabla = 1.0$ and ℓ_2 regularisation strength $\lambda_{\ell_2} = 0.1$. As evaluation metrics, we use FID (Heusel et al., 2018) and Inception Score (Salimans et al., 2016) using the same evaluation regime as in (Ho et al., 2020). To select hyperparameters and track performance during training, we use FID evaluated on a subset of 1024 images from a training set of CIFAR10.

For CIFAR10, we use random flip data augmentation.

In DMMD we have two sets of hyperparameters, one is used for training in Algorithm 1 and one is used for sampling in Algorithm 2. During training, we fix the sampling parameters and always use these to select the best set of training time hyperparameters. We use $\eta = 0.1$ gradient flow learning rate, $T = 10$ number of noise levels, $N_{\text{p}} = 200$ number of noisy particles, $N_{\text{s}} = 5$ number of gradient flow steps per noise level, $t_{\text{min}} = 0.001$ and $t_{\text{max}} = 1 - 0.001$. We use a batch of 400 clean particles during training. For hyperparameters, we do a grid search for $\lambda_\nabla \in \{0, 0.001, 0.01, 0.1, 1.0, 10.0\}$, for $\lambda_{\ell_2} \in \{0, 0.001, 0.01, 0.1, 1.0, 10.0\}$, for dropout rate $\{0, 0.1, 0.2, 0.3\}$, for batch size $\{16, 32, 64\}$. To train the model, we use the same optimization procedure as in (Ho et al., 2020), notably Adam (Kingma & Ba, 2017) optimizer with a learning rate 0.0001. We also swept over the the dimensionality of the output layer $32, 64, 128$, such that each of four feature vectors got the equal dimension. Moreover, we swept over the number of channels for U-Net $\{32, 64, 128\}$ (the original one was 32) and we found that 128 gave us the best empirical results.

After having selected the training-time hyperparameters and having trained the model, we run a sweep for the sampling time hyperparameters over $\eta \in \{1, 0.5, 0.1, 0.04, 0.01\}$, $T \in \{1, 5, 10, 50\}$, $N_{\text{s}} \in \{1, 5, 10, 50\}$, $t_{\text{min}} \in \{0.001, 0.01, 0.1, 0.2\}$, $t_{\text{max}} \in \{0.9, 1 - 0.001\}$. We found that the best hyperparameters for DMMD were $\eta = 0.1$, $N_{\text{s}} = 10$, $T = 10$, $t_{\text{min}} = 0.1$ and $t_{\text{max}} = 0.9$. On top of that, we ran a variant for DMMD with $T = 50$ and $N_{\text{s}} = 5$.

For a-DMMD method, we used the same pretrained discriminator as for DMMD but we did an additional sweep over sampling time hyperparameters, because in principle these could be different. We found that the best hyperparameters for a-DMMD are $\eta = 0.04$, $t_{\text{min}} = 0.2$, $t_{\text{max}} = 0.9$, $T = 5$, $N_{\text{s}} = 10$.

For the denoising step, see Table 2, for DMMD-e, we used 2 steps of DMMD gradient flow with a higher learning rate $\eta^\star = 0.5$ with $t_{\text{max}} = 0.1$ and $t_{\text{min}} = 0.001$. For a-DMMD-e, we used 2 steps of DMMD gradient flow with a higher learning rate of $\eta^\star = 0.5$ with $t_{\text{max}} = 0.2$ and $t_{\text{min}} = 0.001$. For a-DMMD-e, we used 2 steps of DMMD gradient flow with a higher learning rate of $\eta^\star = 0.1$ with

$t_{\text{max}} = 0.2$ and $t_{\text{min}} = 0.001$. The only parameter we swept over in this experiment was this higher learning rate $\eta^{\star}$.

After having found the best hyperparameters for sampling, we run the evaluation to compute FID on the whole CIFAR10 dataset using the same regime as described in (Ho et al., 2020).

For MMD-GAN experiment, we use the same discriminator as for DMMD but on top of that we train a generator using the same U-net architecture as for DMMD with an exception that we do not use the 4 levels of features. We use a higher gradient penalty of $\lambda_{\nabla} = 10$ and the same ℓ_2 penalty $\lambda_{\ell_2} = 0.1$. We use a batch size of $B = 64$ and the same learning rate as for DMMD. We use a dropout of 0.2. We train MMD-GAN for 250000 iterations. For each generator update, we do 5 discriminator updates, following (Brock et al., 2019).

For MMD-GAN-Flow experiment, we take the pretrained discriminator from MMD-GAN and run a gradient flow of type (4) on it, starting from a random noise sampled from a Gaussian. We swept over different parameters such as learning rate η and number of iterations N_{iter}. We found that none of our parameters led to any reasonable performance. The results in Table 1 are reported using $\eta = 0.1$ and $N_{\text{iter}} = 100$.

F.1 Additional datasets

We study performance of DMMD on additional datasets, MNIST (Lecun et al., 1998), on CELEB-A (64x64 (Liu et al., 2015) and on LSUN-Church (64x64) (Yu et al., 2016). For MNIST and CELEB-A, we use the same training/test split as well as the evaluation protocol as in (Franceschi et al., 2023). For LSUN-Church, For LSUN Church, we compute FID on 50000 samples similar to DDPM (Ho et al., 2020). For MNIST, we used the same hyperparameters during training and sampling as for CIFAR-10 with NFE=100, see Appendix F. For CELEB-A and LSUN, we ran a sweep over $\lambda_{\ell_2} \in \{0, 0.001, 0.01, 0.1, 1.0, 10.0\}$ and found that $\ell_2 = 0.001$ led to the best results. For sampling, we used the same parameters as for CIFAR-10 with NFE=100. The reported results in Table 4 are given with NFE=100.

F.1.1 Results on CELEB-A, LSUN-Church and MNIST

Besides CIFAR-10, we study the performance of DMMD on MNIST (Lecun et al., 1998), CELEB-A (64x64 (Liu et al., 2015) and LSUN-Church (64x64) (Yu et al., 2016). For MNIST and CELEB-A, we consider the same splits and evaluation regime as in (Franceschi et al., 2023). For LSUN Church, the splits and the evaluation regime are taken from (Ho et al., 2020). For more details, see Appendix F.1. The results are provided in Table 4. In addition to DMMD, we report the performance of *Discriminator flow* baseline from (Franceschi et al., 2023) with numbers taken from the corresponding paper. We see that DMMD performance is significantly better compared to the discriminator flow, which is consistent with our findings on CIFAR-10. The corresponding samples are provided in Appendix I.2.

Table 4: **Unconditional image generation on additional datasets**. The metric used is FID. The number of gradient flow steps for DMMD is 100.

Dataset	MMD-GAN	DDPM	DMMD	Disc. flow (Franceschi et al., 2023)
MNIST	7.0	1.94	3.0	4.0
CELEB-A 12.1	6.72	8.3	41.0	
LSUN	8.4	3.84	6.1	-

F.2 D-KALE-flow details

We study performance of D-KALE-flow on CIFAR10. We use the same architectural setting as in DMMD with the only difference of adding an additional mapping $\alpha(t;\psi)$ from noise level to D dimensional feature vector, which is represented by a 2 layer MLP with hidden dimensionality of 64 and GELU activation function. We use batch size $B = 256$, dropout rate equal to 0.3. For sampling time parameters during training, we use $\eta = 0.5$, total number of noise levels $T = 20$, and number of steps per noise level $N_s = 5$. At training, we sweep over RKHS regularization $\lambda \in \{0, 1, 10, 100, 500, 1000, 2000\}$, gradient penalty $\lambda_\nabla \in \{0, 0.1, 1.0, 10.0, 50.0, 100.0, 250.0, 500.0, 1000.0\}$, l2 penalty in $\{0, 0.1, 0.01, 0.001\}$.

F.3 Number of particles ablation

Number of particles. In Table 5 we report performance of DMMD depending on the number of particles N_p at sampling time. As expected as the number of particles increases, the FID score decreases, but the overall performance is sensitive to the number of particles. This motivates the approximate sampling procedure from Section 5.

Table 5: **Number of particles ablation**, FIDs on CIFAR10.

$N_\mathrm{p} = 50$	$N_\mathrm{p} = 100$	$N_\mathrm{p} = 200$
9.76	8.55	8.31

G Performance vs. Number of gradient flow steps trade-off

Here, we provide a table showing the dependence of the performance of DMMD on number of total DMMD gradient flow steps, which we call NFE. The NFE is the total number of gradient flow iterations, which equals to $N_s T$, where N_s is the number of steps per noise level and T is the number of noise levels. By default, we use the gradient flow learning rate $\eta = 0.1$, see (9). We also found that as we increase the number of total gradient flow steps, it was sometimes beneficial to use a smaller learning rate, $\eta = 0.05$. Results are given in Table 6. We see that as we increase NFE, the FID improves up to a point (NFE = 250). After NFE=250, we do not see a further improvement. Moreover, as we noticed in our experiments, increasing the total compute at sampling time might require readjusting the gradient flow learning rate.

Table 6: Dependence of the FID on CIFAR-10 on the total number of gradient flow steps (NFE). η is the gradient flow learning rate, see (9).

Total number of steps (NFE)	FID
$10 (\eta = 0.1)$	377.5
$50 (\eta = 0.1)$	36.4
$100 (\eta = 0.1)$	8.5
$250 (\eta = 0.1)$	12.1
$250 (\eta = 0.05)$	7.74
$500 (\eta = 0.05)$	8.6
$1000 (\eta = 0.05)$	9.1

G.1 Results with f-divergence

We study performance of D-KALE-Flow described in Section D and Appendix E, in the setting of unconditional image generation for CIFAR-10. We compare against a GAN baseline which uses the KALE divergence in the discriminator, but has adversarially trained generator. More details are described in Appendix E and Appendix F.2. The results are given in Table 7. We see that unlike with DMMD, D-KALE-Flow achieves worse performance than corresponding KALE-GAN - indicating that the inductive bias provided by the generator may be more helpful in this case - this is a topic for future study.

G.2 Compute resources for image experiments

For all the experiments, we used $A100$ GPUs with 40 GB of memory. To train DMMD for $250k$ steps, we needed to run training for around 24 hours. The total hyperparameter sweep for DMMD

Table 7: **Unconditional image generation on CIFAR-10** with KALE-divergence. The number of gradient flow steps is 100.

Method	FID	Inception score
D-KALE-Flow	15.8	8.5
KALE-GAN	12.7	8.7

required 36 runs to figure out regularization constants, 12 runs to figure out batch size and dropout rate and then 3 runs to figure out the dimensionality of the U-Net and the same 3 runs where the features of the U-Net were coming only from the last layer. This required 54 runs in total.

Running inference on small subset of CIFAR-10 required around 2 minutes of GPU time, and we ran full grid search to select best sampling time parameters, which is around 1080 values. We did this sweep for DMMD and $a-\mathrm{DMMD}$. For $\mathrm{DMMD}-e$, we additionally swept over higher learning rate at the second stage which required 5 more runs. For $a-\mathrm{DMMD}-e$ and $a-\mathrm{DMMD}-a$, we swept over learning rates at second stage which required 10 more runs. After having found the best parameters, we run sampling with the best parameters on full CIFAR-10 which takes about 1 hour for $NFE = 100$.

For additional datasets, for $MNIST$ we used the same best parameters as for CIFAR-10, which required one run only since we saw very good performance out of the box. For CELEB-A and LSUN, we ran an additional sweep over regularization strength which required 6 training runs per dataset and 2 additional runs for sampling the whole datasets.

For $\mathrm{MMD}-GAN$, the training runs were faster, by around 2-x factor. We did a grid search over the regularization strengths which took 36 training runs and 12 runs to figure out batch size and drop-out rate.

For DKALE-flow, the experiment was as fast as $\mathrm{MMD}-GAN$ and we ran a grid search with 67 runs for regularization and 4 runs for dropout. The same was done for $\mathrm{DKALE}-GAN$.

H Optimal kernel with Gaussian MMD

In this section, we prove the results of Section 3. We consider the following unnormalized Gaussian kernel

$$k_\alpha(x, y) = \alpha^{-d} \exp[-\|x-y\|^2/(2\alpha^2)].$$

For any $\mu \in \mathbb{R}^d$ and $\sigma > 0$ we denote $\pi_{\mu,\sigma}$ the Gaussian distribution with mean μ and covariance matrix $\sigma^2 \mathrm{Id}$. We denote MMD^2_α the MMD^2 associated with k_α. More precisely for any $\mu_1, \mu_2 \in \mathbb{R}^d$ and $\sigma_1, \sigma_2 > 0$ we have

$$\mathrm{MMD}^2_\alpha(\pi_{\mu_1,\sigma_1}, \pi_{\mu_2,\sigma_2}) = \mathbb{E}_{\pi_{\mu_1,\sigma_1} \otimes \pi_{\mu_1,\sigma_1}}[k_\alpha(X, X')] - 2\mathbb{E}_{\pi_{\mu_1,\sigma_1} \otimes \pi_{\mu_2,\sigma_2}}[k_\alpha(X, Y)] + \mathbb{E}_{\pi_{\mu_2,\sigma_2} \otimes \pi_{\mu_2,\sigma_2}}[k_\alpha(Y, Y')]. \tag{23}$$

In this section we prove the following result.

Proposition H.1. *For any $\mu_0 \in \mathbb{R}^d$ and $\sigma > 0$, let $\alpha^\star$ be given by*

$$\alpha^\star = \mathrm{argmax}_{\alpha \geq 0} \|\nabla_{\mu_0} \mathrm{MMD}^2_\alpha(\pi_{0,\sigma}, \pi_{\mu_0,\sigma})\|.$$

Then, we have that

$$\alpha^\star = \mathrm{ReLU}(\|\mu_0\|^2/(d+2) - 2\sigma^2)^{1/2}. \tag{24}$$

Before proving Proposition H.1, let us provide some insights on the result. The quantity $\|\nabla_{\mu_0} \mathrm{MMD}^2_\alpha(\pi_{0,\sigma}, \pi_{\mu_0,\sigma})\|$ represents how much the mean of the Gaussian $\pi_{\mu_0,\sigma}$ is displaced when considering a flow on the mean of the Gaussian w.r.t. MMD^2_α. Intuitively, we aim for $\|\nabla_{\mu_0} \mathrm{MMD}^2_\alpha(\pi_{0,\sigma}, \pi_{\mu_0,\sigma})\|$ to be as large as possible as this represents the *maximum displacement possible*. Hence, this justifies our goal of maximizing $\|\nabla_{\mu_0} \mathrm{MMD}^2_\alpha(\pi_{0,\sigma}, \pi_{\mu_0,\sigma})\|$ with respect to the width parameter α.

We show that the optimal width $\alpha^\star$ has a closed form given by (24). It is notable that, assuming that $\sigma > 0$ is fixed, this quantity depends on $\|\mu_0\|$, i.e. how far the modes of the two distributions are.

This observation justifies our approach of following an *adaptive* MMD flow at inference time. Finally, we observe that there exists a threshold, i.e. $\|\mu_0\|^2/(d+2) = 2\sigma^2$ for which lower values of $\|\mu_0\|$ still yield $\alpha^\star = 0$. This phase transition behavior is also observed in our experiments.

We define $\mathrm{D}(\alpha, \sigma, \mu_0, \mu_1)$ for any $\alpha, \sigma > 0$ and $\mu_0, \mu_1 \in \mathbb{R}^d$ given by

$$\mathrm{D}(\alpha, \sigma, \mu_0, \mu_1) = \textstyle\int_{\mathbb{R}^d \times \mathbb{R}^d} k_\alpha(x, y) \mathrm{d}\pi_{\mu_0,\sigma}(x) \mathrm{d}\pi_{\mu_1,\sigma}(y).$$

Proposition H.2. *For any $\alpha, \sigma > 0$ and $\mu_0, \mu_1 \in \mathbb{R}^d$ we have*

$$\begin{aligned}\mathrm{D}(\alpha, \sigma, \mu_0, \mu_1) = [\alpha^2\sigma^2(1/\kappa^2 + 1/\alpha^2)]^{-d/2} \exp[&\|\hat{\mu}_0\|^2/(2\kappa^2) + \|\hat{\mu}_1\|^2/(2\kappa^2) \\ &- \langle \hat{\mu}_0, \hat{\mu}_1 \rangle/\alpha^2 - \|\mu_0\|^2/(2\sigma^2) - \|\mu_1\|^2/(2\sigma^2)],\end{aligned}$$

with

$$\begin{aligned}\hat{\mu}_1 &= (\alpha^2\mu_1 + \kappa^2\mu_0)/(\kappa^2 + \alpha^2), \\ \hat{\mu}_0 &= (\alpha^2\mu_0 + \kappa^2\mu_1)/(\kappa^2 + \alpha^2),\end{aligned}$$

where $\kappa = (1/\sigma^2 + 1/\alpha^2)^{-1/2}$.

Proof. In what follows, we start by computing $\mathrm{D}(\alpha, \sigma, \mu_0, \mu_1)$ for any $\alpha, \sigma > 0$ and $\mu_0, \mu_1 \in \mathbb{R}^d$ given by

$$\begin{aligned}\mathrm{D}(\alpha, \sigma, \mu_0, \mu_1) &= \textstyle\int_{\mathbb{R}^d \times \mathbb{R}^d} k_\alpha(x, y) \mathrm{d}\pi_{\mu_0,\sigma}(x) \mathrm{d}\pi_{\mu_1,\sigma}(y) \\ &= 1/(2\pi\sigma^2\alpha)^d \textstyle\int_{\mathbb{R}^d \times \mathbb{R}^d} \exp[-\|x - y\|^2/(2\alpha^2)] \exp[-\|x - \mu_0\|^2/(2\sigma^2)] \exp[-\|y - \mu_1\|^2/(2\sigma^2)] \mathrm{d}x\mathrm{d}y \\ &= 1/(2\pi\sigma^2\alpha)^d \textstyle\int_{\mathbb{R}^d \times \mathbb{R}^d} \exp[-\|x - y\|^2/(2\alpha^2) - \|x - \mu_0\|^2/(2\sigma^2) - \|y - \mu_1\|^2/(2\sigma^2)] \mathrm{d}x\mathrm{d}y.\end{aligned}$$

In what follows, we denote $\kappa = (1/\sigma^2 + 1/\alpha^2)^{-1/2}$. We have

$$\mathrm{D}(\alpha, \sigma, \mu_0, \mu_1) = C(\mu_0, \mu_1) \textstyle\int_{\mathbb{R}^d \times \mathbb{R}^d} \exp[-\|x\|^2/(2\kappa^2) - \|y\|^2/(2\kappa^2) + \langle x, y \rangle/\alpha^2 + \langle x, \mu_0 \rangle/\sigma^2 + \langle y, \mu_1 \rangle/\sigma^2] \mathrm{d}x\mathrm{d}y,$$

with $C(\mu_0, \mu_1) = 1/(2\pi\sigma^2\alpha)^d \exp[-\|\mu_0\|^2/(2\sigma^2) - \|\mu_1\|^2/(2\sigma^2)]$. In what follows, we denote $\mathrm{P}(x, y)$ the second-order polynomial given by

$$\mathrm{P}(x, y) = \|x\|^2/(2\kappa^2) + \|y\|^2/(2\kappa^2) - \langle x, y \rangle/\alpha^2 - \langle x, \mu_0 \rangle/\sigma^2 - \langle y, \mu_1 \rangle/\sigma^2.$$

Note that we have

$$\mathrm{D}(\alpha, \sigma, \mu_0, \mu_1) = C(\mu_0, \mu_1) \textstyle\int_{\mathbb{R}^d \times \mathbb{R}^d} \exp[-\mathrm{P}(x, y)] \mathrm{d}x\mathrm{d}y. \tag{25}$$

Next, for any $\hat{\mu}_0, \hat{\mu}_1 \in \mathbb{R}^d$, we consider $\mathrm{Q}(x, y)$ given by

$$\begin{aligned}\mathrm{Q}(x, y) &= \|x - \hat{\mu}_0\|^2/(2\kappa^2) + \|y - \hat{\mu}_1\|^2/(2\kappa^2) - \langle x - \hat{\mu}_0, y - \hat{\mu}_1 \rangle/\alpha^2 \\ &= \|x\|^2/(2\kappa^2) + \|\hat{\mu}_0\|^2/(2\kappa^2) + \|y\|^2/(2\kappa^2) + \|\hat{\mu}_1\|^2/(2\kappa^2) - \langle x, \hat{\mu}_0 \rangle/\kappa^2 - \langle y, \hat{\mu}_1 \rangle/\kappa^2 - \langle x - \hat{\mu}_0, y - \hat{\mu}_1 \rangle/\alpha^2 \\ &= \|x\|^2/(2\kappa^2) + \|\hat{\mu}_0\|^2/(2\kappa^2) + \|y\|^2/(2\kappa^2) + \|\hat{\mu}_1\|^2/(2\kappa^2) - \langle x, \hat{\mu}_0 \rangle/\kappa^2 - \langle y, \hat{\mu}_1 \rangle/\kappa^2 \\ &\quad - \langle x, y \rangle/\alpha^2 - \langle \hat{\mu}_0, \hat{\mu}_1 \rangle/\alpha^2 + \langle y, \hat{\mu}_0 \rangle/\alpha^2 + \langle x, \hat{\mu}_1 \rangle/\alpha^2 \\ &= \mathrm{P}(x, y) + \|\hat{\mu}_0\|^2/(2\kappa^2) + \|\hat{\mu}_1\|^2/(2\kappa^2) - \langle x, \hat{\mu}_0 \rangle/\kappa^2 - \langle y, \hat{\mu}_1 \rangle/\kappa^2 + \langle x, \mu_0 \rangle/\sigma^2 + \langle y, \mu_1 \rangle/\sigma^2 \\ &\quad - \langle \hat{\mu}_0, \hat{\mu}_1 \rangle/\alpha^2 + \langle y, \hat{\mu}_0 \rangle/\alpha^2 + \langle x, \hat{\mu}_1 \rangle/\alpha^2 \\ &= \mathrm{P}(x, y) + \|\hat{\mu}_0\|^2/(2\kappa^2) + \|\hat{\mu}_1\|^2/(2\kappa^2) - \langle \hat{\mu}_0, \hat{\mu}_1 \rangle/\alpha^2 \\ &\quad + \langle x, \mu_0/\sigma^2 - \hat{\mu}_0/\kappa^2 + \hat{\mu}_1/\alpha^2 \rangle + \langle y, \mu_1/\sigma^2 - \hat{\mu}_1/\kappa^2 + \hat{\mu}_0/\alpha^2 \rangle.\end{aligned}$$

In what follows, we set $\hat{\mu}_0, \hat{\mu}_1$ such that

$$\begin{aligned}\mu_1/\sigma^2 &= \hat{\mu}_1/\kappa^2 - \hat{\mu}_0/\alpha^2, \\ \mu_0/\sigma^2 &= \hat{\mu}_0/\kappa^2 - \hat{\mu}_1/\alpha^2.\end{aligned}$$

We get that

$$\begin{aligned}\hat{\mu}_1 &= (\mu_1/(\sigma^2\kappa^2) + \mu_0/(\sigma^2\alpha^2))/(1/\kappa^4 - 1/\alpha^4), \\ \hat{\mu}_0 &= (\mu_0/(\sigma^2\kappa^2) + \mu_1/(\sigma^2\alpha^2))/(1/\kappa^4 - 1/\alpha^4).\end{aligned}$$

We have that

$$\sigma^2(1/\kappa^4 - 1/\alpha^4) = \sigma^2(1/\sigma^4 + 2/(\sigma^2\alpha^2)) = 1/\sigma^2 + 2/\alpha^2 = 1/\kappa^2 + 1/\alpha^2. \tag{26}$$

Therefore, we get that

$$\hat{\mu}_1 = (\mu_1/\kappa^2 + \mu_0/\alpha^2)/(1/\kappa^2 + 1/\alpha^2),$$
$$\hat{\mu}_0 = (\mu_0/\kappa^2 + \mu_1/\alpha^2)/(1/\kappa^2 + 1/\alpha^2).$$

Finally, we get that

$$\hat{\mu}_1 = (\alpha^2\mu_1 + \kappa^2\mu_0)/(\kappa^2 + \alpha^2),$$
$$\hat{\mu}_0 = (\alpha^2\mu_0 + \kappa^2\mu_1)/(\kappa^2 + \alpha^2).$$

With this choice, we get that

$$\mathrm{P}(x, y) = \mathrm{Q}(x, y) - \|\hat{\mu}_0\|^2/(2\kappa^2) - \|\hat{\mu}_1\|^2/(2\kappa^2) + \langle \hat{\mu}_0, \hat{\mu}_1 \rangle/\alpha^2 \tag{27}$$

We also have that for any $x, y \in \mathbb{R}^d$

$$\mathrm{Q}(x, y) = (1/2) \begin{pmatrix} x - \hat{\mu}_0 \\ y - \hat{\mu}_1 \end{pmatrix}^\top \begin{pmatrix} \mathrm{Id}/\kappa^2 & -\mathrm{Id}/\alpha^2 \\ -\mathrm{Id}/\alpha^2 & \mathrm{Id}/\kappa^2 \end{pmatrix} \begin{pmatrix} x - \hat{\mu}_0 \\ y - \hat{\mu}_1 \end{pmatrix}$$

Using this result we have that

$$\textstyle\int_{\mathbb{R}^d \times \mathbb{R}^d} \exp[-\mathrm{Q}(x, y)] = (2\pi)^d \det(\Sigma^{-1})^{-1/2}, \tag{28}$$

with

$$\Sigma^{-1} = \begin{pmatrix} \mathrm{Id}/\kappa^2 & -\mathrm{Id}/\alpha^2 \\ -\mathrm{Id}/\alpha^2 & \mathrm{Id}/\kappa^2 \end{pmatrix}.$$

Using (26), we get that

$$\det(\Sigma^{-1}) = [(1/\sigma^2)(1/\kappa^2 + 1/\alpha^2)]^d.$$

Combining this result and (28) we get that

$$\textstyle\int_{\mathbb{R}^d \times \mathbb{R}^d} \exp[-\mathrm{Q}(x, y)] = (2\pi)^d [(1/\sigma^2)(1/\kappa^2 + 1/\alpha^2)]^{-d/2}.$$

Combining this result, (27) and (25) we get that

$$\mathrm{D}(\alpha, \sigma, \mu_0, \mu_1) = C(\mu_0, \mu_1) \exp[\|\hat{\mu}_0\|^2/(2\kappa^2) + \|\hat{\mu}_1\|^2/(2\kappa^2) - \langle \hat{\mu}_0, \hat{\mu}_1 \rangle/\alpha^2](2\pi)^d [(1/\sigma^2)(1/\kappa^2 + 1/\alpha^2)]^{-d/2}.$$

Therefore, we get that

$$\mathrm{D}(\alpha, \sigma, \mu_0, \mu_1) = [\alpha^2\sigma^2(1/\kappa^2 + 1/\alpha^2)]^{-d/2} \exp[\|\hat{\mu}_0\|^2/(2\kappa^2) + \|\hat{\mu}_1\|^2/(2\kappa^2)$$
$$- \langle \hat{\mu}_0, \hat{\mu}_1 \rangle/\alpha^2 - \|\mu_0\|^2/(2\sigma^2) - \|\mu_1\|^2/(2\sigma^2)].$$

□

We investigate two special cases of Proposition H.2.

First, we show that if $\mu_0 = \mu_1$ then $\mathrm{D}(\alpha, \sigma, \mu_0, \mu_0)$ does not depend on μ_0.

Proposition H.3. *For any $\alpha, \sigma > 0$ and $\mu_0 \in \mathbb{R}^d$ we have $\mathrm{D}(\alpha, \sigma, \mu_0, \mu_0) = (\alpha^2 + 2\sigma^2)^{-d/2}$.*

Proof. We have that $\hat{\mu}_0 = \hat{\mu}_1 = \mu_1 = \mu_0$ in Proposition H.2. In addition, we have that

$$(1/2\kappa^2) + (1/2\kappa^2) - 1/\alpha^2 - 1/(2\sigma^2) - 1/(2\sigma^2) = 0.$$

Therefore, we have that

$$\exp[\|\hat{\mu}_0\|^2/(2\kappa^2) + \|\hat{\mu}_1\|^2/(2\kappa^2) - \langle \hat{\mu}_0, \hat{\mu}_1 \rangle/\alpha^2 - \|\mu_0\|^2/(2\sigma^2) - \|\mu_1\|^2/(2\sigma^2)] = 1,$$

which concludes the proof upon using that $1/\kappa^2 = 1/\alpha^2 + 1/\sigma^2$. □

Proposition H.3 might seem surprising at first but in fact it simply highlights the fact that when trying to differentiate a Gaussian measure with itself, the result is independent of the location of the Gaussian and only depends on its scale. Then, we study the case where $\mu_1 = 0$.

Proposition H.4. *For any $\alpha, \sigma > 0$ and $\mu_0 \in \mathbb{R}^d$ we have*

$$\mathrm{D}(\alpha, \sigma, \mu_0, 0) = (\alpha^2 + 2\sigma^2)^{-d/2} \exp[-\|\mu_0\|^2/(2(\alpha^2 + 2\sigma^2))].$$

Proof. First, we have that

$$\hat{\mu}_0 = \alpha^2/(\kappa^2 + \alpha^2)^2 \mu_0, \qquad \hat{\mu}_1 = \kappa^2/(\kappa^2 + \alpha^2)^2 \mu_0.$$

Therefore, we get that

$$\mathrm{D}(\alpha, \sigma, \mu_0, 0) = [\sigma^2(1/\kappa^2 + 1/\alpha^2)]^{d/2} \exp[(1/2)\{(\alpha^4/\kappa^2 - \kappa^2)/(\kappa^2 + \alpha^2) - 1/\sigma^2\}\|\mu_0\|^2]$$

Using (26) we get that

$$\alpha^4/\kappa^2 - \kappa^2 = \alpha^2(\alpha^2 + \kappa^2)/\sigma^2.$$

Therefore, we get that

$$(\alpha^4/\kappa^2 - \kappa^2)/(\kappa^2 + \alpha^2) - 1/\sigma^2 = (\alpha^2/(\alpha^2 + \kappa^2) - 1)/\sigma^2 = -1/(\alpha^2(1 + 2\sigma^2/\alpha^2)),$$

which concludes the proof. □

Using Proposition H.3, Proposition H.4 and definition (23), we have the following result.

Proposition H.5. *For any $\alpha, \sigma > 0$ and $\mu_0 \in \mathbb{R}^d$ we have*

$$\mathrm{MMD}^2(\pi_{0,\sigma}, \pi_{\mu_0,\sigma}) = 2(\alpha^2 + 2\sigma^2)^{-d/2}(1 - \exp[-\|\mu_0\|^2/(2(\alpha^2 + 2\sigma^2))]).$$

In addition, we have

$$\nabla_{\mu_0} \mathrm{MMD}^2(\pi_{0,\sigma}, \pi_{\mu_0,\sigma}) = -2(\alpha^2 + 2\sigma^2)^{-d/2-1} \exp[-\|\mu_0\|^2/(2(\alpha^2 + 2\sigma^2))]\mu_0.$$

Finally, we have the following proposition.

Proposition H.6. *For any $\mu_0 \in \mathbb{R}^d$ and $\sigma > 0$ let $\alpha^\star$ be given by*

$$\alpha^\star = \mathrm{argmax}_{\alpha \geq 0} \|\nabla_{\mu_0} \mathrm{MMD}^2(\pi_{0,\sigma}, \pi_{\mu_0,\sigma})\|.$$

Then, we have that

$$\alpha^\star = \mathrm{ReLU}(\|\mu_0\|^2/(d+2) - 2\sigma^2)^{1/2}.$$

Proof. Let $\sigma > 0$ and $\mu_0 \in \mathbb{R}^d$. First, using Proposition H.5, we have that for

$$\|\nabla_{\mu_0} \mathrm{MMD}^2(\pi_{0,\sigma}, \pi_{\mu_0,\sigma})\|^2 = 4\alpha^{2d}\|\mu_0\|^2(\alpha^2 + 2\sigma^2)^{-d-2} \exp[-\|\mu_0\|^2/(\alpha^2 + 2\sigma^2)].$$

Next, we study the function $\mathrm{f} : [0, t_0] \to \mathbb{R}$ given for any $t \in [0, t_0]$ by

$$\mathrm{f}(t) = t^{d+2} \exp[-t\|\mu_0\|^2],$$

with $t_0 = 1/(2\sigma^2)$. We have that

$$\mathrm{f}'(t) = t^{d+1} \exp[-t\|\mu_0\|^2]((d+2) - \|\mu_0\|^2 t).$$

We then consider two cases. First, if $t_0 \leq (d+2)/\|\mu_0\|^2$, i.e. $\sigma^2 \leq \|\mu_0\|^2/(2(d+2))$, then f is increasing on $[0, t_0]$ and we have that f is maximum if $t = t_0$. Hence, if $\sigma^2 \leq \|\mu_0\|^2/(2(d+2))$, we have that $\alpha^\star = 0$. Second, if $t_0 \leq (d+2)/\|\mu_0\|^2$, i.e. $\sigma^2 \leq \|\mu_0\|^2/(2(d+2))$ then f is increasing on $[0, t^\star]$, non-increasing on $[t^\star, t_0]$ with $t^\star = (d+2)/\|\mu_0\|^2$ and we have that f is maximum if $t = t^\star$. Hence, if $\sigma^2 \geq \|\mu_0\|^2/(2(d+2))$, we have that $\alpha^\star = (\|\mu_0\|^2/(d+2) - 2\sigma^2)^{1/2}$, which concludes the proof. □

H.1 Phase transition behaviour

I Image generation samples

I.1 CIFAR10 samples

Samples from DMMD with NFE=100 and NFE=250 are given in Figure 4. Samples from DMMD with NFE=100 and from a-DMMD with NFE=50 are given in Figure 5.

I.2 Additional datasets samples

Samples for MNIST are given in Figure 6, for CELEB-A (64x64) are given in Figure 7 and for LSUN Church (64x64) are given in Figure 8.

Figure 3: Evolution of the norm of the mean μ_t of the Gaussian distribution $\pi_{\mu_t,\sigma}$ according to a gradient flow on the mean μ_t w.r.t. MMD_{α_t}. In the *adaptive* case α_t is given by Proposition 3.1 while in the *non adaptive* case, $\alpha_t = \alpha_0 = 1$. In our experiment we consider $d = 1$ and $\sigma = 1$, for illustration purposes.

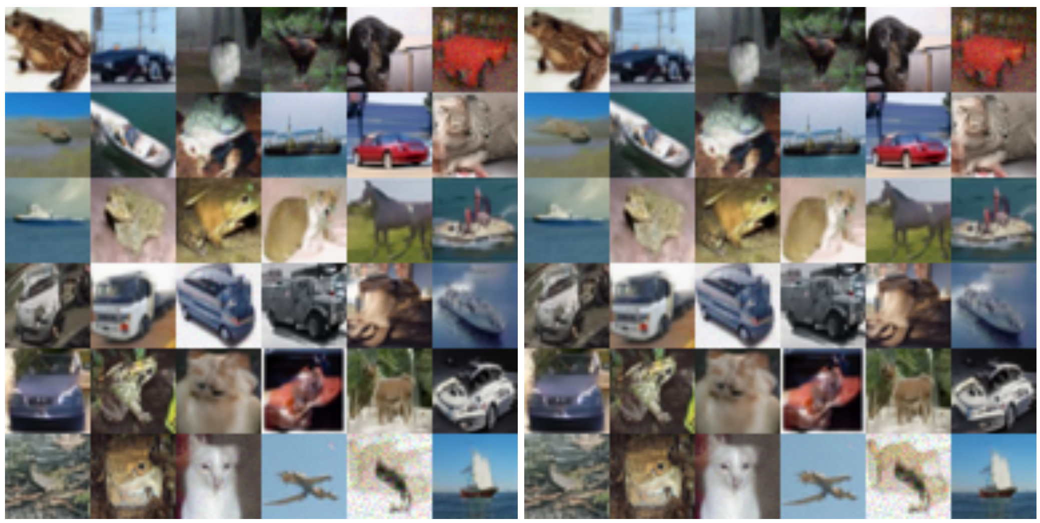

Figure 4: CIFAR-10 samples from DMMD with NFE=250 on the left and with NFE=100 on the right

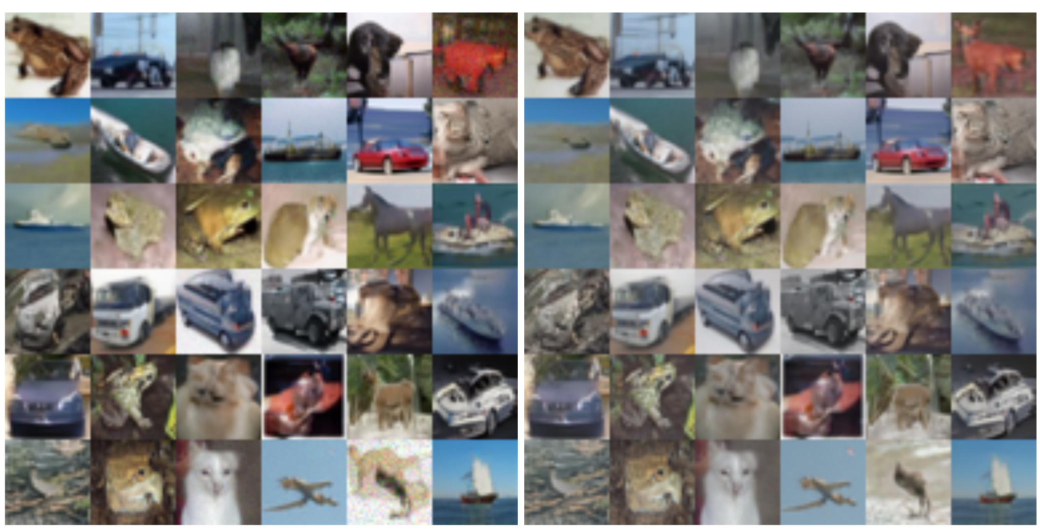

Figure 5: CIFAR-10 samples from DMMD with NFE=100 on the left and samples from the a-DMMD-e with NFE=50 on the right

Figure 6: DMMD samples for MNIST.

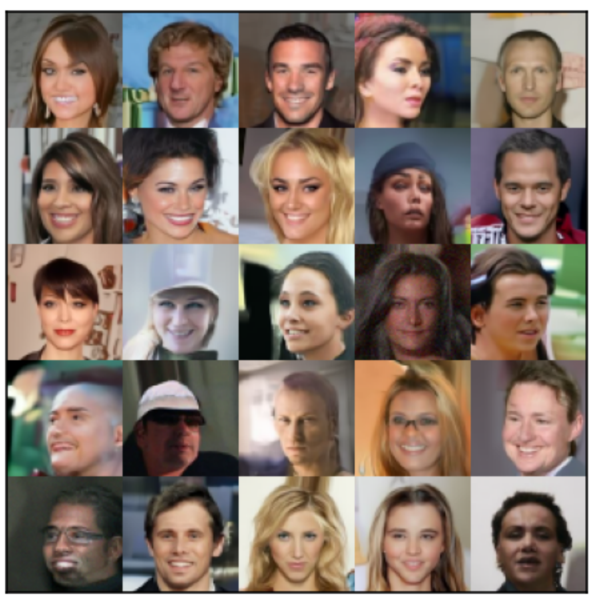

Figure 7: DMMD samples for CELEB-A (64x64).

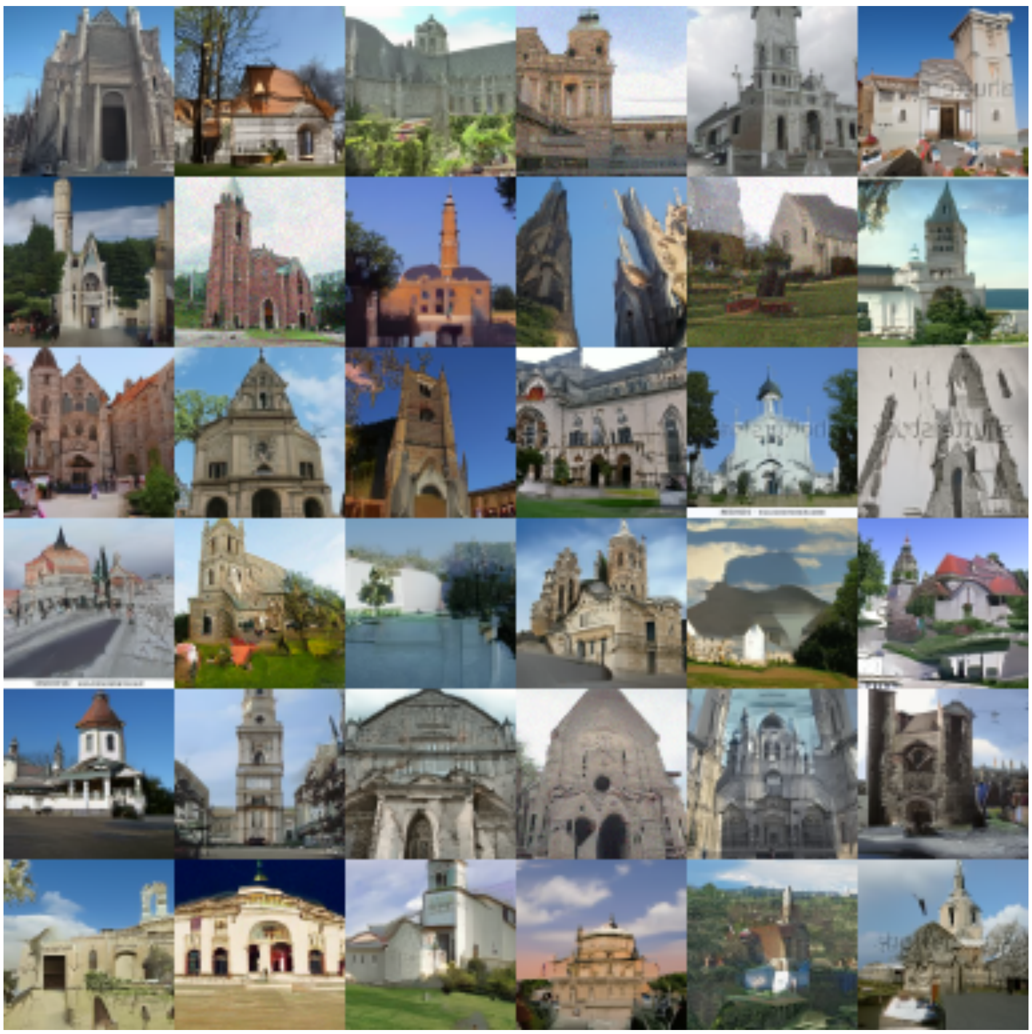

Figure 8: DMMD samples for LSUN Church (64x64).