Training generative neural networks using Maximum Mean Discrepancy

Vlad Shakhuro

Faculty of Computer Science, NRU HSE

21 October 2016

Equality of probability distributions

Let x and y be two random variables:
$x \sim p(x),\ y \sim q(y)$

Given observations
$X = \{x_1, \ldots, x_N\},\quad Y = \{y_1, \ldots, y_M\}$

we'd like to decide wheter
$p \stackrel{?}{=} q$

Maximum Mean Discrepancy

$$\mathrm{MMD}[\mathcal{F}, \mathrm{p}, \mathrm{q}] = \sup_{\mathrm{f} \in \mathcal{F}} (\mathrm{E}_{\mathrm{p}} \mathrm{f}(\mathrm{x}) - \mathrm{E}_{\mathrm{q}} \mathrm{f}(\mathrm{y}))$$

Theorem

Let $\mathcal{F}$ be a unit ball in a universal RKHS $\mathcal{H}$, defined on the compact metric space $\mathcal{X}$ with associated kernel $\mathrm{k}(\cdot, \cdot)$. Then $\mathrm{MMD}[\mathcal{F}, \mathrm{p}, \mathrm{q}] = 0$ iff $\mathrm{p} = \mathrm{q}$.

Gretton et al. A Kernel Two-Sample Test. JMLR 2012

Maximum Mean Discrepancy

Given that $\mathrm{k}(\mathrm{x}, \mathrm{y}) = \langle \phi(\mathrm{x}), \phi(\mathrm{y}) \rangle$, we can obtain:

$$\mathrm{MMD}^2[\mathcal{H}, \mathrm{p}, \mathrm{q}] = \|\mathrm{E}_{\mathrm{p}} \phi(\mathrm{x}) - \mathrm{E}_{\mathrm{q}} \phi(\mathrm{y})\|^2 =$$

$$\mathrm{E}_{\mathrm{pp}} \langle \phi(\mathrm{x}), \phi(\mathrm{x}') \rangle - \mathrm{E}_{\mathrm{pq}} \langle \phi(\mathrm{x}), \phi(\mathrm{y}) \rangle + \mathrm{E}_{\mathrm{qq}} \langle \phi(\mathrm{y}), \phi(\mathrm{y}') \rangle =$$

$$\mathrm{E}_{\mathrm{pp}} \mathrm{k}(\mathrm{x}, \mathrm{x}') - \mathrm{E}_{\mathrm{pq}} \mathrm{k}(\mathrm{x}, \mathrm{y}) + \mathrm{E}_{\mathrm{qq}} \mathrm{k}(\mathrm{y}, \mathrm{y}')$$

Maximum Mean Discrepancy

Given that $k(x, y) = \langle \phi(x), \phi(y) \rangle$, we can obtain:

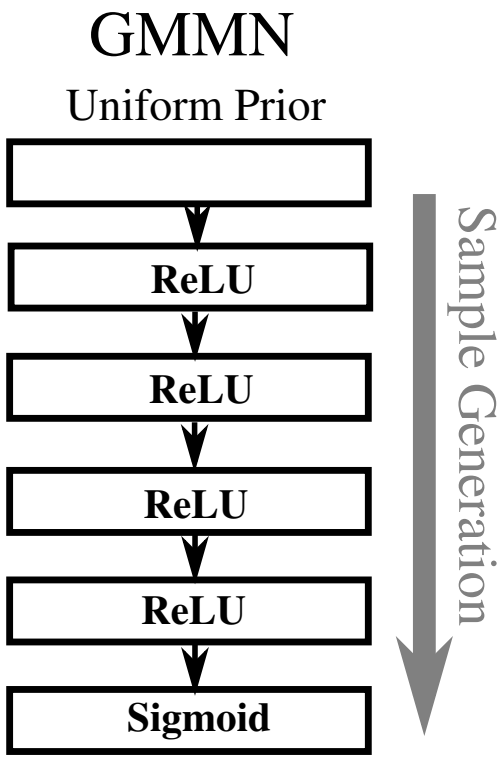
$$MMD^2[\mathcal{H}, p, q] = \|E_p \phi(x) - E_q \phi(y)\|^2 =$$
$$E_{pp} \langle \phi(x), \phi(x') \rangle - E_{pq} \langle \phi(x), \phi(y) \rangle + E_{qq} \langle \phi(y), \phi(y') \rangle =$$
$$E_{pp} k(x, x') - E_{pq} k(x, y) + E_{qq} k(y, y')$$

Biased estimate for practical use:

$$MMD_b^2[\mathcal{H}, X, Y] =$$

$$\frac{1}{N^2} \sum_{n,n'} k(x_n, x_{n'}) - \frac{2}{MN} \sum_{n,m} k(x_n, y_m) + \frac{1}{M^2} \sum_{m,m'} k(y_m, y_{m'})$$

MMD neural network



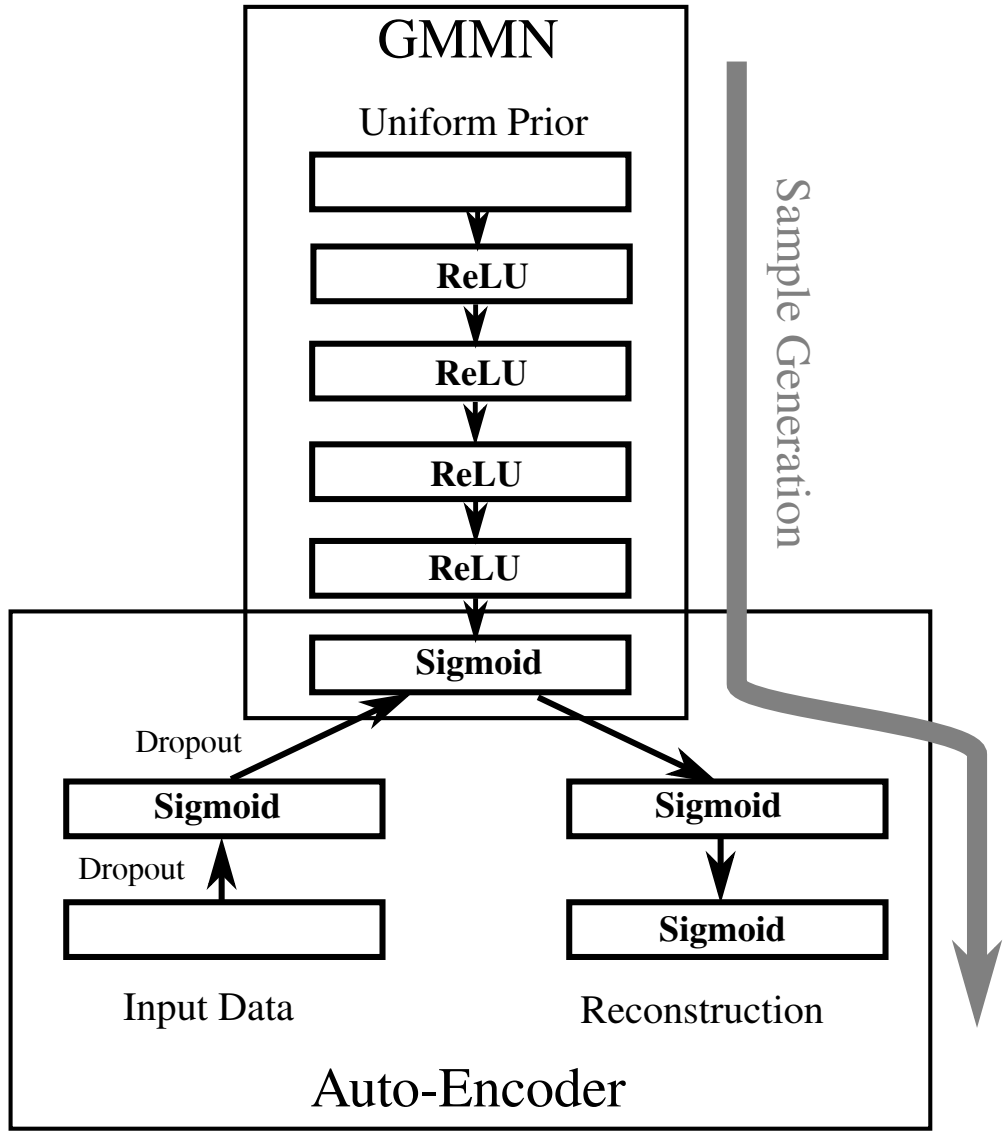
Training:

given dataset $\{x_1^d, \ldots, x_N^d\}$,
prior $p(z)$, neural net $G(z; w)$

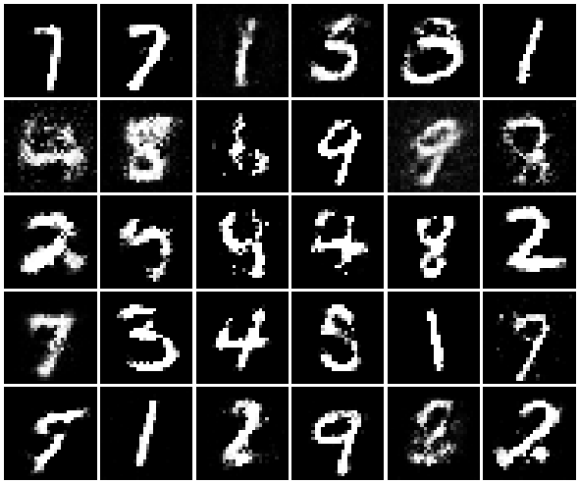
while not stop:
- get a minibatch of data X^d
- generate samples X^s
- compute $\frac{\partial MMD}{\partial w}$ on X^d and X^s
- update w using gradient

Li et al. Generative Moment Matching Networks. JMLR 2015

MMD + Autoencoder

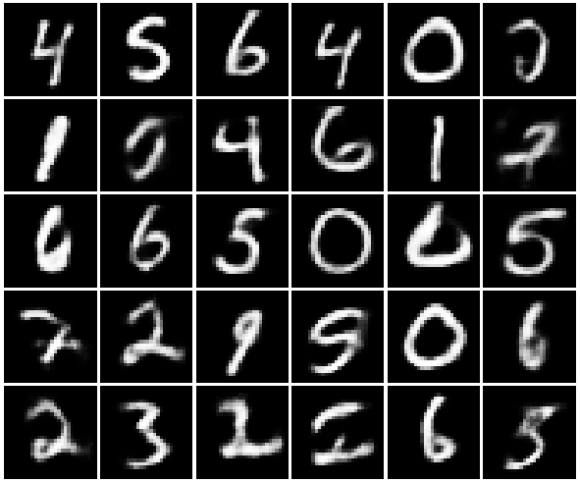


Results



(a) GMMN MNIST samples

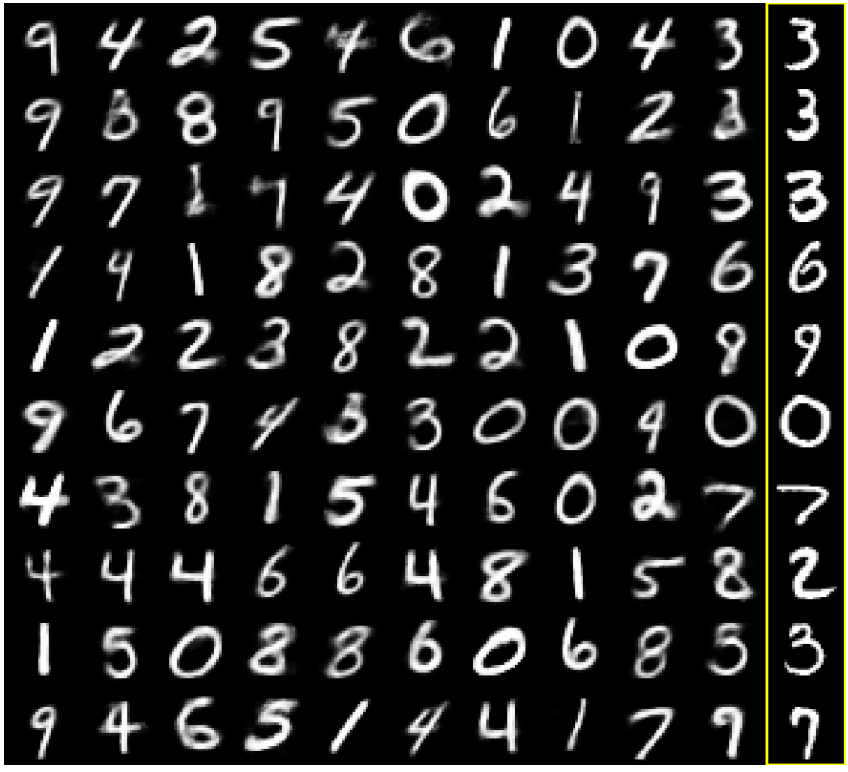
(b) GMMN TFD samples



(c) GMMN+AE MNIST samples

(d) GMMN+AE TFD samples

Results (adversarial autoencoders)



(a) MNIST samples (8-D Gaussian)

(b) TFD samples (15-D Gaussian)

Makhzani et al. Adversarial Autoencoders. ICLR 2016

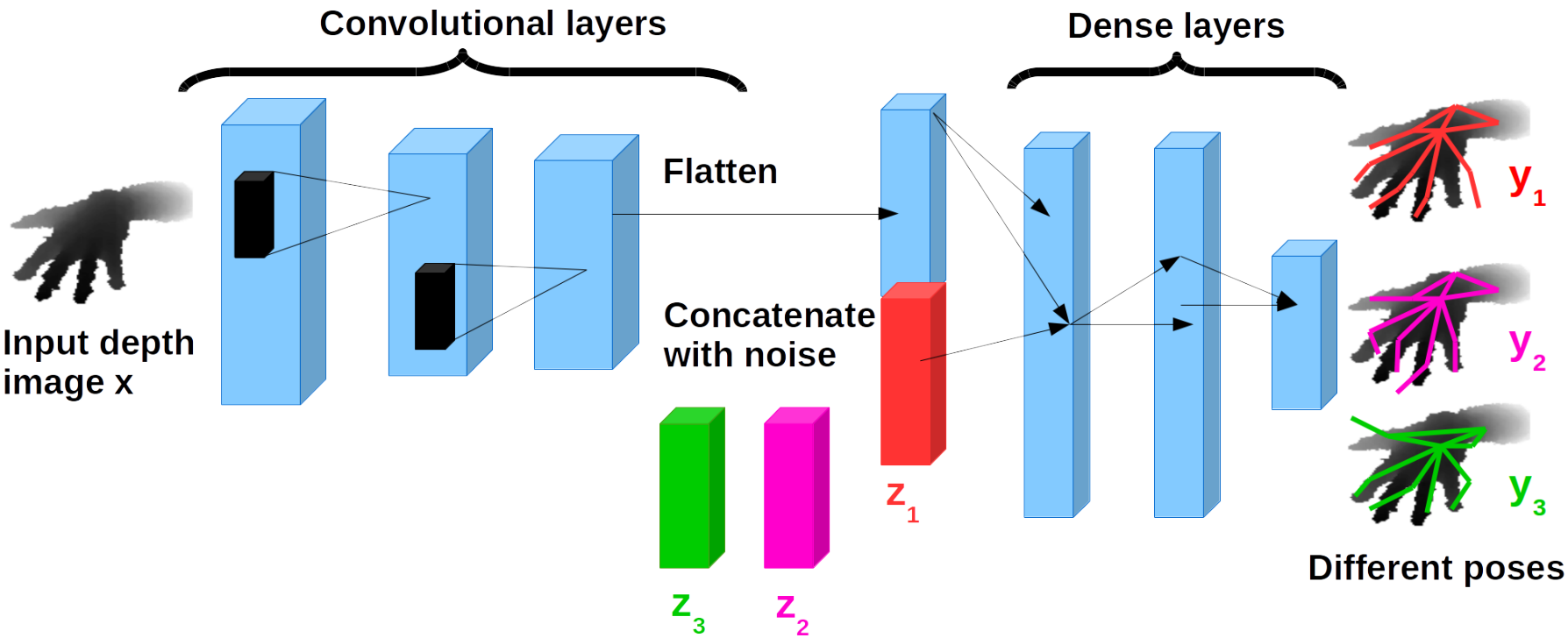
Generative net for posterior distribution

Given training dataset $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\}$, we'd like to learn $\mathbf{p}(\mathbf{y}|\mathbf{x})$

Bouchacourt et al. DISCO Nets: DISsimilarity COefficient Networks. NIPS 2016

Generative net for posterior distribution

Given training dataset $\{(x_1, y_1), \ldots, (x_m, y_m)\}$, we'd like to learn $p(y|x)$



Bouchacourt et al. DISCO Nets: DISsimilarity COefficient Networks. NIPS 2016

Objective function

We want $\mathrm{p_G}(\mathrm{y}|\mathrm{x})$ be similar to $\mathrm{p_T}(\mathrm{y}|\mathrm{x})$ evaluated with respect to task-specific loss function $\Delta(\mathrm{y}, \mathrm{y}')$

$\mathrm{DIV}_\Delta(\mathrm{p_T}, \mathrm{p_G}) =$
$\int_\mathrm{x} \int_\mathrm{y} \int_{\mathrm{y}'} \Delta(\mathrm{y}, \mathrm{y}') \mathrm{p_T}(\mathrm{y}|\mathrm{x}) \mathrm{p_G}(\mathrm{y}'|\mathrm{x}) \mathrm{p_T}(\mathrm{x}) \mathrm{d}\mathrm{y} \mathrm{d}\mathrm{y}' \mathrm{d}\mathrm{x}$

Rao. Diversity and dissimilarity coefficients: A unified approach. Theoretical Population Biology, 1982

Objective function

We want $\mathrm{p_G}(\mathrm{y}|\mathrm{x})$ be similar to $\mathrm{p_T}(\mathrm{y}|\mathrm{x})$ evaluated with respect to task-specific loss function $\Delta(\mathrm{y}, \mathrm{y}')$

$$\mathrm{DIV}_\Delta(\mathrm{p_T}, \mathrm{p_G}) = \int_\mathrm{x} \int_\mathrm{y} \int_{\mathrm{y}'} \Delta(\mathrm{y}, \mathrm{y}') \mathrm{p_T}(\mathrm{y}|\mathrm{x}) \mathrm{p_G}(\mathrm{y}'|\mathrm{x}) \mathrm{p_T}(\mathrm{x}) \mathrm{dy} \mathrm{dy}' \mathrm{dx}$$

$$\mathrm{DISC}_\Delta(\mathrm{p_T}, \mathrm{p_G}) = \mathrm{DIV}_\Delta(\mathrm{p_T}, \mathrm{p_G}) - \gamma \mathrm{DIV}_\Delta(\mathrm{p_G}, \mathrm{p_G}) - (1-\gamma) \mathrm{DIV}_\Delta(\mathrm{p_T}, \mathrm{p_T})$$

Rao. Diversity and dissimilarity coefficients: A unified approach. Theoretical Population Biology, 1982

Objective function

We want $p_G(y|x)$ be similar to $p_T(y|x)$ evaluated with respect to task-specific loss function $\Delta(y, y')$

$$DIV_\Delta(p_T, p_G) = \int_x \int_y \int_{y'} \Delta(y, y') p_T(y|x) p_G(y'|x) p_T(x) dy dy' dx$$

$$DISC_\Delta(p_T, p_G) = DIV_\Delta(p_T, p_G) - \gamma DIV_\Delta(p_G, p_G) - (1-\gamma) DIV_\Delta(p_T, p_T)$$

$$F = DIV_\Delta(p_T, p_G) - \gamma DIV_\Delta(p_G, p_G)$$

Rao. Diversity and dissimilarity coefficients: A unified approach. Theoretical Population Biology, 1982

Unbiased estimates

$$\mathrm{DIV}_\Delta(p_T, p_G) = \frac{1}{NK} \sum_{n=1}^{N} \sum_{k=1}^{K} \Delta(y_n, G(z_k, x_n))$$

$$\mathrm{DIV}_\Delta(p_G, p_G) = \frac{1}{NK(K-1)} \sum_{n=1}^{N} \sum_{k,k' \neq k} \Delta(G(z_k, x_n), G(z_{k'}, x_n))$$

Gneiting, Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. Journal of the American Statistical Association, 2007. Theorem 5

Unbiased estimates

$$\mathrm{DIV}_\Delta(p_T, p_G) = \frac{1}{NK} \sum_{n=1}^{N} \sum_{k=1}^{K} \Delta(y_n, G(z_k, x_n))$$

$$\mathrm{DIV}_\Delta(p_G, p_G) = \frac{1}{NK(K-1)} \sum_{n=1}^{N} \sum_{k,k' \neq k} \Delta(G(z_k, x_n), G(z_{k'}, x_n))$$

If $\gamma = \frac{1}{2}$ and $\Delta(y, y') = \|y - y'\|_2$, then

$$F = \frac{1}{N} \sum_n \left(\frac{1}{K} \sum_k \|y_n - G(z_k, x_n)\|_2 - \frac{1}{2K(K-1)} \sum_{k,k' \neq k} \|G(z_{k'}, x_n) - G(z_k, x_n)\|_2 \right)$$

F is minimised iff $p_T = p_G$

Gneiting, Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. Journal of the American Statistical Association, 2007. Theorem 5

Metrics

Table 2: *Evaluation metrics. N is the number of testing example pairs $(\boldsymbol{x}_n, \boldsymbol{y}_n)$ and $\boldsymbol{y}_{\Delta_{metric},n}$ is the prediction, specific to the metric, for the n^{th} input example $\boldsymbol{x}_n$. J is the number of joints of the hand.*

Shorthand & Definition	Δ_{metric}	Formula
ProbLoss: Probabilistic Loss of K sampled poses.	$\Delta_{\text{ProbLoss}} = \|\|.\|\|_2^{\beta}$	$\widehat{F}(\Delta_{\text{ProbLoss}}, \boldsymbol{\theta})$ with $\gamma = 0.5$
MeJEE (Mean Joint Euclidean Error) : per-joint Euclidean distance between the pointwise pose and the goundtruth, averaged by J and N.	$\Delta_{\text{MeJEE}} = \frac{1}{J}\sum_{j=1}^{J} \|\|.\|\|_2^{\text{joint}j}$	$\frac{1}{N}\sum_{n=1}^{N}\frac{1}{J}\sum_{j=1}^{J} \|\|\boldsymbol{y}_n^j - \boldsymbol{y}_{\Delta_{\text{MeJEE}},n}^j\|\|_2^{\text{joint}j}$
MaJEE (Max Joint Euclidean Error) per-joint maximal Euclidean distance between the pointwise pose and the goundtruth averaged by N.	$\Delta_{\text{MaJEE}} = \max_{j\in[1,J]} \|\|.\|\|_2^{\text{joint}j}$	$\frac{1}{N}\sum_{n=1}^{N}\max_{j\in[1,J]} \|\|\boldsymbol{y}_n^j - \boldsymbol{y}_{\Delta_{\text{MaJEE}},n}^j\|\|_2^{\text{joint}j}$
FF(d) (Fraction of Frames) : fraction of test examples that have all predicted joints of the pointwise pose below a given maximum Euclidean distance d in mm from the ground-truth.	$\Delta_{\text{FF}} = -\mathbb{1}_{\max_{j\in[1,J]} \|\|.\|\|_2^{\text{joint}j} \leq d}$ (note the minus sign since we want to maximise FF)	$\frac{1}{N}\sum_{n=1}^{N}\mathbb{1}_{\max_{j\in[1,J]} \|\|\boldsymbol{y}_n^j - \boldsymbol{y}_{\Delta_{\text{FF}},n}^j\|\|_2^{\text{joint}j} \leq d}$

Getting point prediction from samples

For fixed $\mathbf{x}$, we generate K samples and get one that minimizes expected task-specific loss Δ, estimated using generated samples

$$\mathbf{y} = \underset{\mathrm{k}=1\ldots\mathrm{K}}{\arg\min} \sum_{\mathrm{k}'=1}^{\mathrm{K}} \Delta(\mathbf{y}_{\mathrm{k}}, \mathbf{y}_{\mathrm{k}'})$$

Results

Table 3: *Metrics values on the test set ± SEM. Best performances in bold.*

Model	ProbLoss (mm)	MeJEE (mm)	MaJEE (mm)	FF (80mm)
$\text{BASE}_{\beta=1,\sigma=1}$	103.8±0.627	25.2±0.152	52.7±0.290	86.040
$\text{BASE}_{\beta=1,\sigma=5}$	99.3±0.620	25.5±0.151	52.9±0.289	85.773
$\text{BASE}_{\beta=1,\sigma=10}$	96.3±0.612	25.7±0.149	53.2±0.288	85.664
$\text{DISCO}_{\beta=1,\gamma=0.5}$	**83.8 ±0.503**	**20.9±0.124**	**45.1±0.246**	**94.438**

Table 4: *Metrics values on the test set ± SEM for cGAN and DISCO Nets. Best performances in bold.*

Model	ProbLoss (mm)	MeJEE (mm)	MaJEE (mm)	FF (80mm)
cGAN	442.7±0.513	109.8±0.128	201.4±0.320	0.000
$\text{cGAN}_{\text{init, fixed}}$	128.9±0.480	31.8±0.117	64.3±0.230	78.454
$\text{DISCO}_{\beta=1,\gamma=0.5}$	**83.8 ±0.503**	**20.9±0.124**	**45.1±0.246**	**94.438**

Table 5: *DISCO Nets compared to state-of-the-art performances ± SEM.*

Model	MeJEE (mm)	MaJEE (mm)	FF (80mm)
NYU-Prior	20.7±0.150	44.8±0.289	91.190
NYU-Prior-Refined	19.7±0.157	44.7±0.327	88.148
NYU-Init	27.4±0.152	55.4±0.265	86.537
NYU-Feedback	16.0±0.096	36.1±0.208	97.334
$\text{DISCO}_{\beta=1,\gamma=0.5}$	20.7±0.121	45.1±0.246	93.250

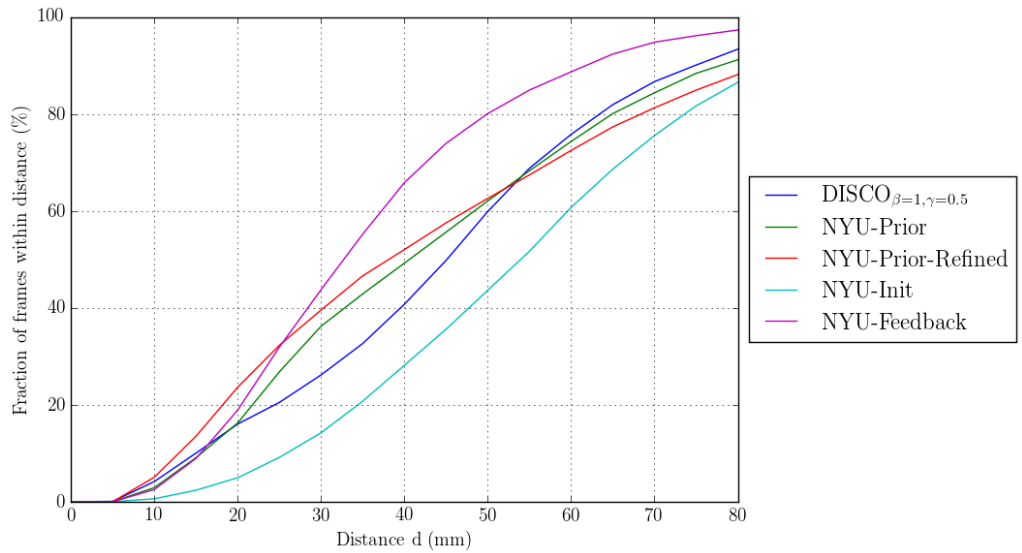


Figure 4: *Fractions of frames within distance d in mm (by 5 mm). Best viewed in color.*

Summary

MMD is a theoretically grounded loss function for training generative models

MMD is easier and faster to implement and train than GANs

Further work on kernels for conditional prediction is required