UNIVERSITY OF WATERLOO

Multivariate Triangular Quantile Maps for Novelty Detection

Jingjing Wang, Sun Sun, Yaoliang Yu
{jingjing.wang, sun.sun, yaoliang.yu}@uwaterloo.ca

Summary

✓ We extend the univariate quantile to the multivariate setting through increasing triangular maps.

✓ We present a new framework for neural novelty detection, which unifies and extends many existing approaches.

✓ We apply the multiple gradient descent algorithm and obtain an efficient end-to-end implementation.

✓ We perform extensive experiments to confirm the efficacy of our proposed framework.

General Framework for Novelty Detection

Triangular Quantile Map

Let $U \sim \text{Uniform}[0,1]$, and X be any *univariate* random variable. Then:

$$F_X(x) = \Pr(X \le x), \qquad Q_X(u) = F_X^{-1}(u) := \inf\{x : F_X(x) \ge u\} \quad \Longrightarrow \quad Q_X(U) \sim X.$$

Challenge: How to extend the quantile function Q_X to the *multivariate* setting?

Definition. Let $\mathbf{U} \sim \text{Uniform}[0,1]^d$, and $\mathbf{X}$ be any random vector in $\mathbb{R}^d$. We call an **increasing triangular** map $\mathbf{Q} = (Q_1, \ldots, Q_d) : [0,1]^d \to \mathbb{R}^d$ *the* Triangular Quantile Map (TQM) of $\mathbf{X}$ if $\mathbf{Q}(\mathbf{U}) \sim \mathbf{X}$.

Increasing: Q_j is increasing *w.r.t.* the j-th coordinate of the input;

Triangular: Q_j depends only on the first j coordinates of the input;

Significance:

- $\mathbf{Q}$ always exists and is unique (up to ordering of coordinates);
- efficiently computable inverse and Jacobian $\Longrightarrow$ efficient density;
- amenable to neural parameterization (real-NVP, MAF, IAF, NAF, SOS, etc);
- composition friendly (unlike optimal transport approach).

Problem Formulation

Let $\mathbf{X}$ be a random sample from the unknown density p. We learn the feature map $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^m$ and the TQM $\mathbf{Q} : \mathbb{R}^m \to \mathbb{R}^m$ *simultaneously* by minimizing:

$$\min_{\mathbf{f}, \mathbf{Q}} \; \gamma \mathsf{KL}(\mathbf{f}_{\#}p \| \mathbf{Q}_{\#}q) + \lambda \ell(\mathbf{f}) + \zeta g(\mathbf{Q}), \qquad (1)$$

- q: a fixed reference density (e.g., the uniform density over $[0,1]^m$)
- ℓ: loss associated with learning the feature map $\mathbf{f}$
- g: potential constraints on the triangular quantile map $\mathbf{Q}$

Novelty Detection Rules

① **Density Scoring Rule**: novel if $p_{\mathbf{Z}}(\mathbf{z}) = 1/|\mathbf{Q}'(\mathbf{Q}^{-1}(\mathbf{z}))| < \tau$ where $\mathbf{z} = \mathbf{f}(\mathbf{X})$.

② **Quantile Scoring Rule**: novel if $\mathbf{Q}^{-1}(\mathbf{f}(\mathbf{X})) \notin N \subseteq [0,1]^m$ whose (uniform) measure is $1 - \alpha$.

Advantages of Our General Framework:

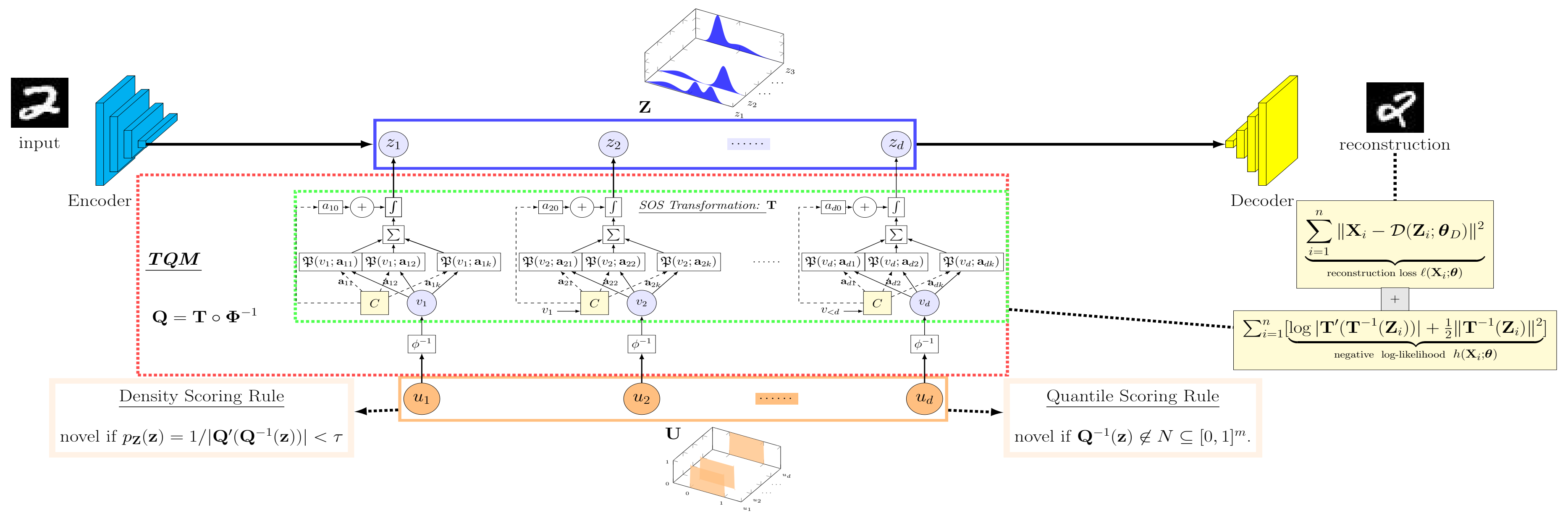
✓ alleviates curse-of-dimensionality by performing dimensionality reduction;

✓ universally consistent and effective by adopting recent flow-based density estimators;

✓ employs the density and quantile scoring rules simultaneously by estimating $\mathbf{Q}$ once;

✓ unifies existing approaches, *e.g.*,*One-class SVM, Support Vector Data, Latent Space Autoregression*.

Estimating TQM Using Deep Networks


input
Encoder
Z
z_1
z_2
z_d
z_3
Decoder
reconstruction
TQM
$\mathbf{Q} = \mathbf{T} \circ \mathbf{\Phi}^{-1}$
SOS Transformation: T
a_{10}
a_{20}
a_{d0}
$\mathfrak{P}(v_1; \mathbf{a}_{11})$
$\mathfrak{P}(v_1; \mathbf{a}_{12})$
$\mathfrak{P}(v_1; \mathbf{a}_{1k})$
$\mathfrak{P}(v_2; \mathbf{a}_{21})$
$\mathfrak{P}(v_2; \mathbf{a}_{22})$
$\mathfrak{P}(v_2; \mathbf{a}_{2k})$
$\mathfrak{P}(v_d; \mathbf{a}_{d1})$
$\mathfrak{P}(v_d; \mathbf{a}_{d2})$
$\mathfrak{P}(v_d; \mathbf{a}_{dk})$
C
v_1
v_2
v_d
$v_{<d}$
ϕ^{-1}
u_1
u_2
u_d
U
$\sum_{i=1}^{n} \|\mathbf{X}_i - \mathcal{D}(\mathbf{Z}_i; \boldsymbol{\theta}_D)\|^2$
reconstruction loss $\ell(\mathbf{X}_i;\boldsymbol{\theta})$
+
$\sum_{i=1}^{n}[\log|\mathbf{T}'(\mathbf{T}^{-1}(\mathbf{Z}_i))| + \frac{1}{2}\|\mathbf{T}^{-1}(\mathbf{Z}_i)\|^2]$
negative log-likelihood $h(\mathbf{X}_i;\boldsymbol{\theta})$
Density Scoring Rule
novel if $p_{\mathbf{Z}}(\mathbf{z}) = 1/|\mathbf{Q}'(\mathbf{Q}^{-1}(\mathbf{z}))| < \tau$
Quantile Scoring Rule
novel if $\mathbf{Q}^{-1}(\mathbf{z}) \notin N \subseteq [0,1]^m$.

Our framework (1) has three components which we implement as follows:

- **Feature Extractor** $\mathbf{f}$: We use a deep autoencoder composed by $\mathbf{Z} = \mathcal{E}(\mathbf{X}; \boldsymbol{\theta}_E)$ and $\hat{\mathbf{X}} = \mathcal{D}(\mathbf{Z}; \boldsymbol{\theta}_D)$, and we use the Euclidean reconstruction loss

$$\ell(\mathbf{f}) = \ell(\boldsymbol{\theta}_E, \boldsymbol{\theta}_D) = \sum_{i=1}^{n} \|\mathbf{X}_i - \hat{\mathbf{X}}_i\|^2. \qquad (2)$$

- **Neural Density Estimator for TQM** $\mathbf{Q} = \mathbf{T} \circ \mathbf{\Phi}^{-1}$:

$\mathbf{\Phi} = (\Phi, \ldots, \Phi)$, where Φ is the CDF of standard univaraite Gaussian;

$\mathbf{T}$ is the TQM implemented by the flow-based density estimation algorithms, e.g., SoS flow;

$\mathsf{KL}(\mathbf{f}_{\#}p \| \mathbf{Q}_{\#}q)$ is approximated empirically using the given sample $\mathbf{X}_1, \ldots, \mathbf{X}_n$ as:

$$\sum_{i=1}^{n} \left[\log|\mathbf{Q}'(\mathbf{Q}^{-1}(\mathbf{f}(\mathbf{X}_i)))| - \log q(\mathbf{Q}^{-1}(\mathbf{f}(\mathbf{X}_i))) \right] \qquad (3)$$

- For example, $g(\mathbf{Q}) = 0$ when using the SoS flow which satisfies the constraints naturally.

Final Objective Function

$$\min_{\boldsymbol{\theta}} \; \sum_{i=1}^{n} (1-\lambda) \Big[\underbrace{\log|\mathbf{T}'(\mathbf{T}^{-1}(\mathbf{Z}_i))| + \tfrac{1}{2}\|\mathbf{T}^{-1}(\mathbf{Z}_i)\|^2}_{\text{negative log-likelihood } h(\mathbf{X}_i;\boldsymbol{\theta})} \Big] + \lambda \underbrace{\|\mathbf{X}_i - \mathcal{D}(\mathbf{Z}_i; \boldsymbol{\theta}_D)\|^2}_{\text{reconstruction loss } \ell(\mathbf{X}_i;\boldsymbol{\theta})} \qquad (4)$$

MGDA for auto-tuning hyperparameter λ

We cast the two competing objectives in (4) as **multi-objective optimization** and implement the Multiple Gradient Descent Algorithm (MGDA) to automatically tune λ in each iteration.

$$\lambda_t = \arg\min_{0 \le \lambda \le 1} \left\| \sum_{i \in I} (1-\lambda) \nabla h(\mathbf{X}_i; \boldsymbol{\theta}_t) + \lambda \nabla \ell(\mathbf{X}_i; \boldsymbol{\theta}_t) \right\|^2 = \min\left\{1, \max\left\{0, \tfrac{\langle \nabla h_I - \nabla \ell_I, \nabla h_I \rangle}{\|\nabla h_I - \nabla \ell_I\|^2}\right\}\right\}, \qquad (5)$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta[(1-\lambda_t)\nabla h_I + \lambda_t \nabla \ell_I], \quad \text{where} \quad I \subseteq \{1, \ldots, n\} \text{ is a minibatch of samples} \qquad (6)$$

Advantages of MGDA:

✓ converges to a **Pareto-stationary solution** under fairly general conditions;

✓ eliminates the need of tuning the hyperparameter λ.

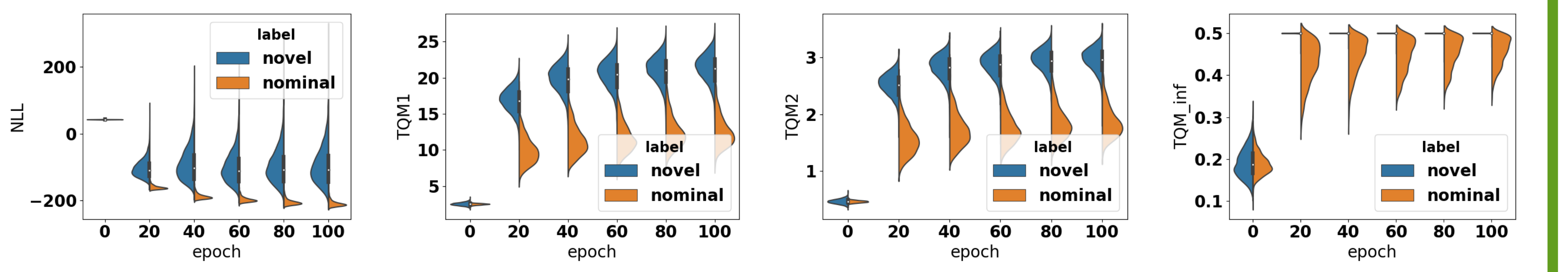
Empirical Results

- **Comparing AUROC of our methods against SOTA on (Fashion) MNIST**:

MNIST											
OC-SVM RAW	OC-SVM CAE	VAE	DAE	LSA	GT	DAGMM	GPND	DSEBM	Ours-NLL	Ours-TQM$_2$	Class
0.995	0.990	0.985	0.982	0.998	0.982	0.500	0.999	0.320	0.995	0.993	0
0.999	0.999	0.997	0.998	0.999	0.893	0.766	0.999	0.987	0.998	0.997	1
0.926	0.919	0.943	0.936	0.923	0.993	0.326	0.980	0.482	0.953	0.948	2
0.936	0.939	0.916	0.929	0.974	0.987	0.319	0.968	0.753	0.963	0.957	3
0.967	0.946	0.945	0.940	0.955	0.993	0.368	0.980	0.696	0.966	0.963	4
0.955	0.936	0.929	0.928	0.966	0.994	0.490	0.987	0.727	0.962	0.960	5
0.987	0.979	0.977	0.982	0.992	0.999	0.515	0.998	0.954	0.992	0.990	6
0.966	0.951	0.975	0.971	0.969	0.966	0.500	0.988	0.911	0.969	0.966	7
0.903	0.896	0.864	0.857	0.935	0.974	0.467	0.929	0.536	0.955	0.951	8
0.962	0.960	0.967	0.974	0.969	0.993	0.813	0.993	0.905	0.977	0.976	9
0.960	0.952	0.950	0.950	0.968	0.977	0.508	**0.982**	0.727	0.973	0.970	avg

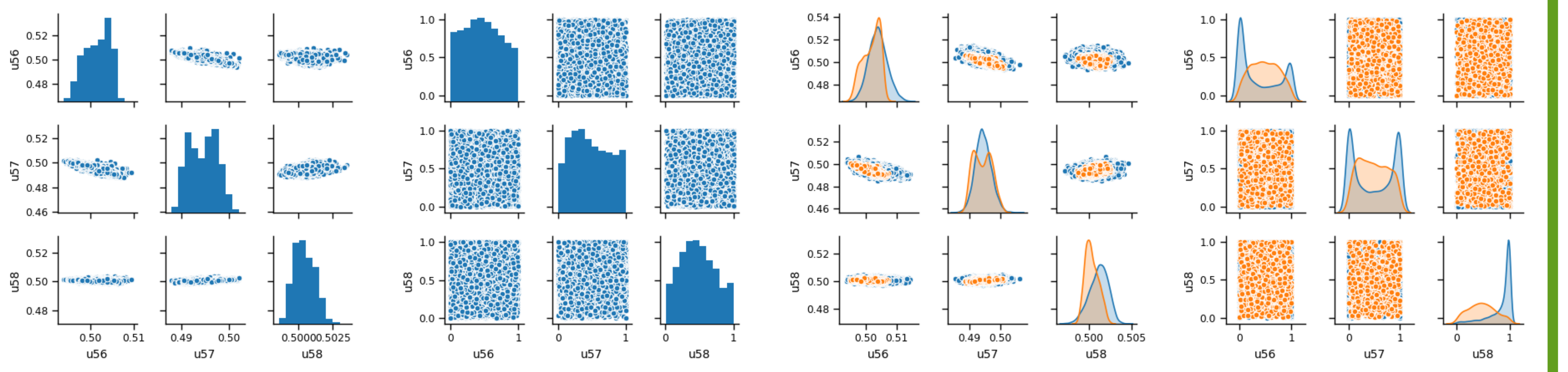
	Fashion-MNIST										
Class	OC-SVM RAW	OC-SVM CAE	VAE	DAE	LSA	GT	DAGMM	GPND	DSEBM	Ours-NLL	Ours-TQM$_2$
0	0.919	0.908	0.874	0.867	0.916	0.903	0.303	0.917	0.891	0.922	0.917
1	0.990	0.987	0.977	0.978	0.983	0.993	0.311	0.983	0.560	0.958	0.950
2	0.894	0.884	0.816	0.808	0.878	0.927	0.475	0.878	0.861	0.899	0.899
3	0.942	0.911	0.912	0.914	0.923	0.906	0.481	0.945	0.903	0.930	0.925
4	0.907	0.913	0.872	0.865	0.897	0.907	0.499	0.906	0.884	0.922	0.921
5	0.918	0.865	0.916	0.921	0.907	0.954	0.413	0.924	0.859	0.894	0.884
6	0.834	0.820	0.738	0.738	0.841	0.832	0.420	0.785	0.782	0.844	0.838
7	0.988	0.984	0.976	0.977	0.977	0.981	0.374	0.984	0.981	0.980	0.972
8	0.903	0.877	0.795	0.782	0.910	0.976	0.518	0.916	0.865	0.945	0.943
9	0.982	0.955	0.965	0.963	0.984	0.994	0.378	0.876	0.967	0.983	0.983
avg	0.928	0.910	0.884	0.881	0.922	**0.937**	0.472	0.911	0.855	0.928	0.923

- **Comparing scoring statistics on MNIST test set** (nominal: 1):



1) NLL: $\log|\mathbf{T}'(\mathbf{T}^{-1}(\tilde{\mathbf{Z}}))| + \frac{1}{2}\|\mathbf{T}^{-1}(\tilde{\mathbf{Z}})\|_2^2$, 2) TQM$_1$: $\|\mathbf{\Phi}(\mathbf{T}^{-1}(\tilde{\mathbf{Z}})) - \frac{1}{2}\|_1$, 3) TQM$_2$: $\|\mathbf{\Phi}(\mathbf{T}^{-1}(\tilde{\mathbf{Z}})) - \frac{1}{2}\|_2$, 4) TQM$_\infty$:$\|\mathbf{\Phi}(\mathbf{T}^{-1}(\tilde{\mathbf{Z}})) - \frac{1}{2}\|_\infty$.

- **Inspecting the training progress**:



1) train at epoch 0; 2) train at epoch 1000; 3) test at epoch 0; 4) test at epoch 1000.

- **Comparing MGDA with two-stage training strategies**:

Fix-: First train the autoencoder, then fix the autoencoder and train the estimation network alone;

Pretrain-: First train the autoencoder, then jointly train the whole framework.

	Fix-NLL	Pretrain-NLL	Ours-NLL	Fix-TQM$_2$	Pretrain-TQM$_2$	Ours-TQM$_2$
0	0.9939	**0.9954**	0.9951	0.9904	**0.9939**	0.9925
1	0.9971	**0.9988**	0.9977	0.9972	**0.9985**	0.9969
2	0.9403	**0.9677**	0.9526	0.9188	**0.9568**	0.9479
3	0.9568	0.9496	**0.9627**	**0.9481**	0.9414	0.9567
4	**0.9703**	0.9445	0.9657	**0.9700**	0.9388	0.9625
5	0.9612	0.9564	**0.9618**	0.9525	0.9486	**0.9601**
6	0.9878	0.9907	**0.9915**	0.9841	0.9881	**0.9895**
7	0.9629	0.9676	**0.9686**	0.9587	0.9656	**0.9660**
8	0.9549	**0.9587**	0.9551	0.9397	**0.9527**	0.9512
9	0.9736	0.9733	**0.9768**	0.9742	0.9641	**0.9756**
avg	0.9699	0.9703	**0.9728**	0.9634	0.9649	**0.9699**

	Thyroid			KDDCUP		
Method	Precision	Recall	F1	Precision	Recall	F1
RAW-OC-SVM *	0.3639	0.4239	0.3887	0.7457	0.8523	0.7954
DSEBM *	0.0404	0.0403	0.0403	0.7369	0.7477	0.7423
DAGMM *	0.4766	0.4834	0.4782	0.9297	0.9442	0.9369
Ours-REC	–	–	–	0.6305	0.6287	0.6296
Ours-NLL	0.7312	0.7312	0.7312	**0.9622**	**0.9622**	**0.9622**
Ours-TQM$_1$	0.5269	0.5269	0.5269	0.9621	0.9621	0.9621
Ours-TQM$_2$	0.5806	0.5806	0.5806	**0.9622**	**0.9622**	**0.9622**
Ours-TQM$_\infty$	**0.7527**	**0.7527**		**0.9622**	**0.9622**	**0.9622**