Tencent AI Lab

# Learning Set Functions Under the Optimal Subset Oracle via Equivariant Variational Inference

Zijing Ou, Tingyang Xu, Qinliang Su, Yingzhen Li, Peilin Zhao, Yatao Bian ✉
Tencent AI Lab
Sun Yat-sen University
Imperial College London

2022-03-11

# Set Function Learning Applications

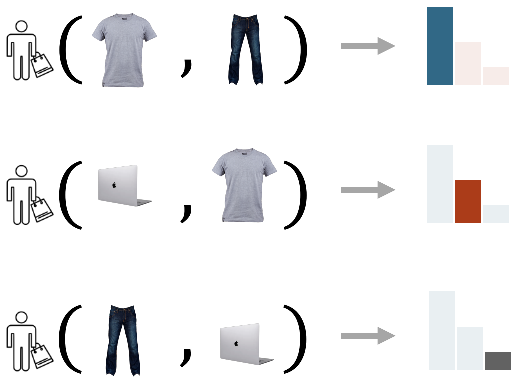
Tencent AI Lab

Database

Customer

Shopping cart

Ground set $V$



Utility function $F_{\theta^*}(\cdot; V)$

Optimal subset $S^*$

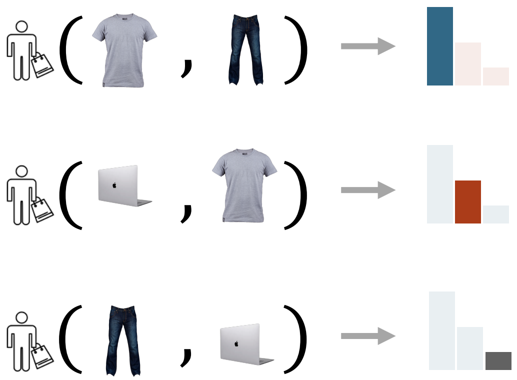
# Set Function Learning Applications

Tencent AI Lab

Database

Customer

Shopping cart



Ground set $V$

Utility function $F_{\theta^*}(\cdot; V)$

Optimal subset $S^*$

**Data Generation Process:**

$$S^* = \underset{S \in 2^V}{\operatorname{argmax}} F_{\theta^*}(S; V)$$

$$\sim \mathbb{p}(S, V) =: \delta_{S=S^*|V}$$

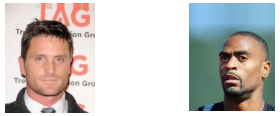
# Set Function Learning Applications



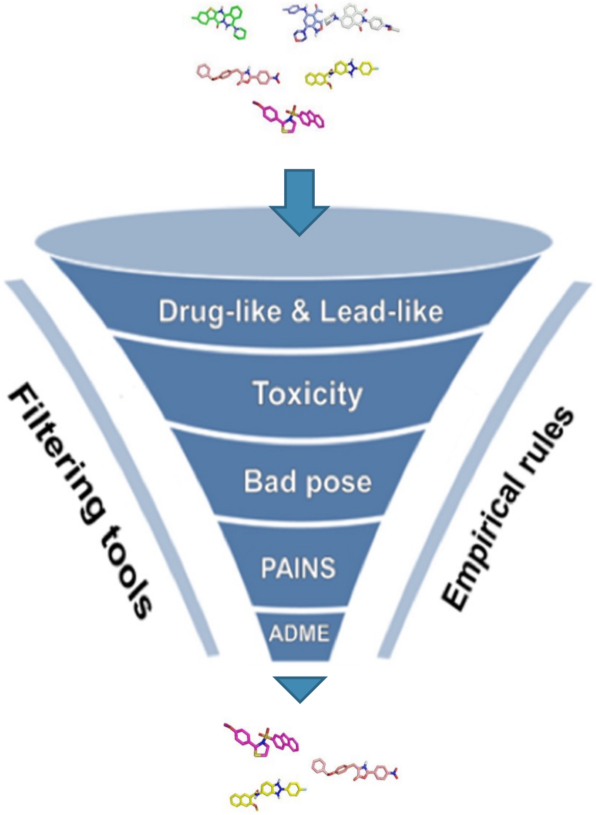
$$S^* = \underset{S \in 2^V}{\operatorname{argmax}} F_{\theta^*}(S; V)$$

## Set anomaly detection

$F_{\theta^*}(S; V) = \sum_{s_i \in S} \mathbb{I}(s_i \ is \ male) - \mathbb{I}(s_i \ is \ femail)$



## Compound selection



# Set Function Learning Function Value (FV) Oracle

Tencent AI Lab

$$S^* = \underset{S \in 2^V}{\operatorname{argmax}} F_{\theta^*}(S; V)$$

**Goal:** Learn a surrogate $F_{\theta}$ to approximate the oracle utility function $F_{\theta^*}$.

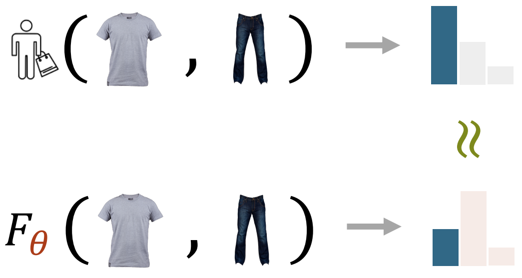
# Set Function Learning Function Value (FV) Oracle

Tencent AI Lab

$$S^* = \underset{S \in 2^V}{\operatorname{argmax}} F_{\theta^*}(S; V)$$

**Goal:** Learn a surrogate $F_\theta$ to approximate the oracle utility function $F_{\theta^*}$.

**Setting 1, FV oracle:**




Matching via empirical risk minimization

$$\theta^* = \min_\theta \sum_i L(F_\theta(S_i; V); F_{\theta^*}(S_i; V))$$

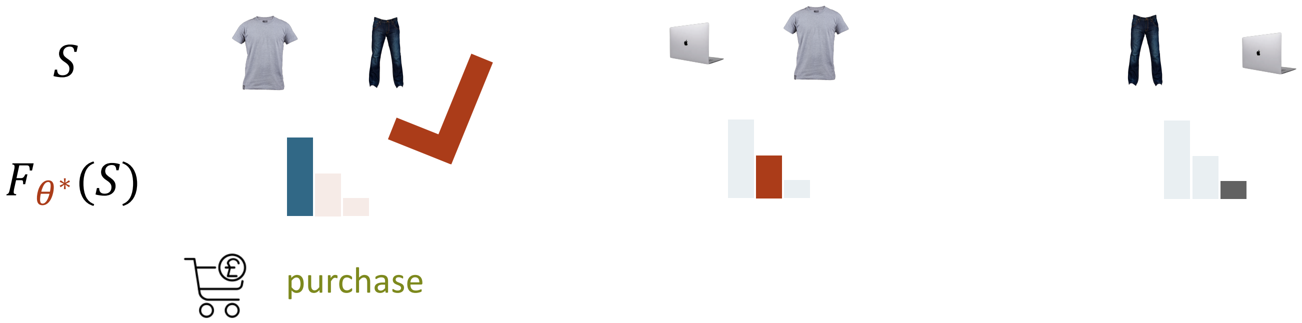
# Set Function Learning Function Value (FV) Oracle

$$S^* = \underset{S \in 2^V}{\operatorname{argmax}} F_{\theta^*}(S; V)$$

**Goal:** Learn a surrogate $F_\theta$ to approximate the oracle utility function $F_{\theta^*}$.

**Setting 1, FV oracle:**

$$\theta^* = \min_{\theta} \sum_i L(F_\theta(S_i; V); F_{\theta^*}(S_i; V))$$



# Set Function Learning Function Value (FV) Oracle

Tencent AI Lab

$$S^* = \underset{S \in 2^V}{\operatorname{argmax}} F_{\theta^*}(S; V)$$

**Goal:** Learn a surrogate $F_\theta$ to approximate the oracle utility function $F_{\theta^*}$.

**Setting 1, FV oracle:**

$$\theta^* = \min_{\theta} \sum_i L(F_\theta(S_i; V); F_{\theta^*}(S_i; V))$$

Curse of amounts of supervision signals 😅

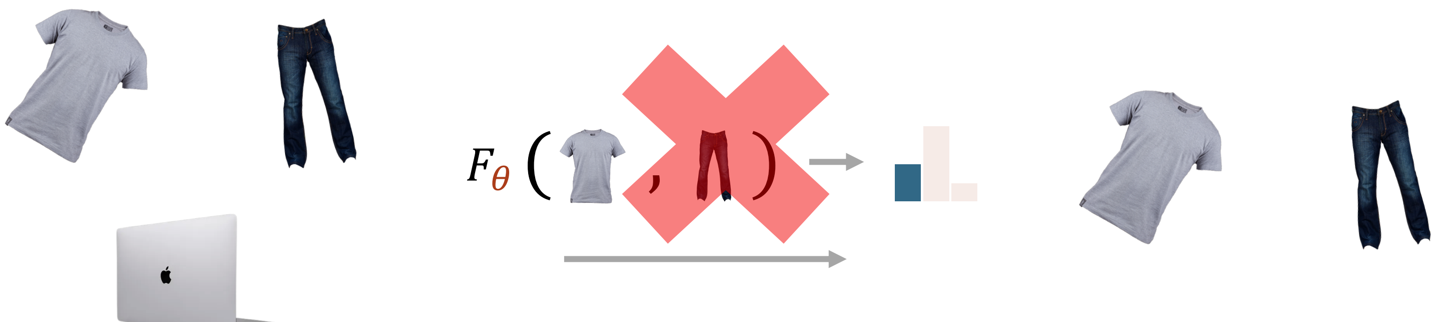
$\Rightarrow$ Training data in form of $\{(S_i, F_{\theta^*}(S_i; V))\}$ for each $V$

# Set Function Learning Optimal Subset (OS) Oracle

$$S^* = \underset{S \in 2^V}{\operatorname{argmax}} F_{\theta^*}(S; V)$$

**Goal:** Learn a surrogate $F_\theta$ to approximate the oracle utility function $F_{\theta^*}$.

**Setting 2, OS oracle:**




Ground set $V$ Directly predict Optimal subset $S^*$

$\Rightarrow$ Training data in form of $\{(S^*, V)\}$

# Learning Set Function Under the OS Oracle

Empirical distribution

$$\underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{\mathbb{P}(S^*,V)}[\log p_\theta(S^*|V)]$$

$$s.t.\, p_\theta(S|V) \propto F_\theta(S;V), \forall S \in 2^V$$

Monotonically grows with the utility function

# Learning Set Function Under the OS Oracle

Empirical distribution

$$\underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{\mathbb{P}(S^*,V)}[\log p_\theta(S^*|V)]$$

$$s.t.\, p_\theta(S|V) \propto F_\theta(S;V), \forall S \in 2^V$$

Monotonically grows with the utility function

**How to construct a proper set mass function $p_\theta(S|V)$?**

# Learning Set Function Under the OS Oracle



$$\underset{\theta}{\operatorname{argmax}}\ \mathbb{E}_{\mathbb{P}(S^*,V)}[\log p_\theta(S^*|V)]$$

$$s.t.\ p_\theta(S|V) \propto F_\theta(S;V), \forall S \in 2^V$$

**Desiderata:**

*Permutation invariance*

$F_\theta($  ,  $)$ = $F_\theta($  ,  $)$

# Learning Set Function Under the OS Oracle

Tencent AI Lab

$$\operatorname*{argmax}_{\theta} \mathbb{E}_{\mathbb{P}(S^*,V)}[\log p_\theta(S^*|V)]$$

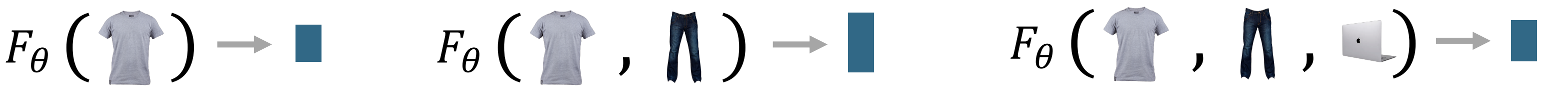
$$s.t.\, p_\theta(S|V) \propto F_\theta(S;V), \forall S \in 2^V$$

**Desiderata:**

*Permutation invariance*



*Varying ground set*



# Learning Set Function Under the OS Oracle



$$\underset{\theta}{\operatorname{argmax}}\ \mathbb{E}_{\mathbb{P}(S^*,V)}[\log p_\theta(S^*|V)]$$

$$s.t.\ p_\theta(S|V) \propto F_\theta(S;V), \forall S \in 2^V$$

**Desiderata:**

*Permutation invariance*



*Varying ground set*



*Differentiability; Minimum prior & Scalability*

# Energy-based Modeling

Tencent AI Lab

$$p_\theta(S|V) = \frac{\exp(F_\theta(S;V))}{Z}$$

$Z \leftarrow$ Partition function $Z \coloneqq \sum_{S\subseteq V} \exp(F_\theta(S;V))$

# Energy-based Modeling

$$p_\theta(S|V) = \frac{\exp(F_\theta(S;V))}{Z}$$

$Z$ ← Partition function $Z \coloneqq \sum_{S \subseteq V} \exp(F_\theta(S;V))$

**EBMs for Minimum Prior:**

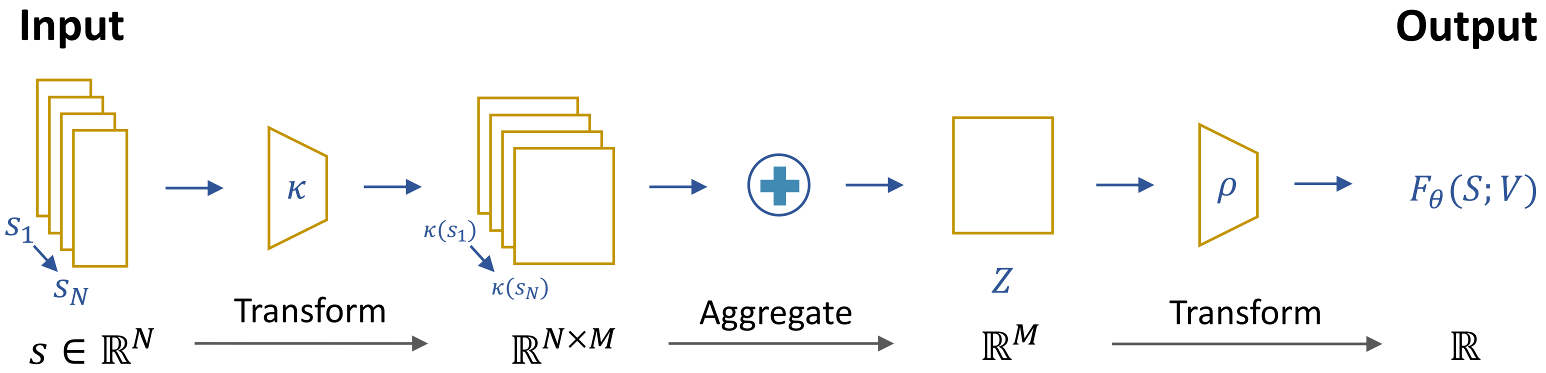
Energy-based modeling has **maximum entropy**

# Energy-based Modeling

$$p_\theta(S|V) = \frac{\exp(F_\theta(S;V))}{Z}$$

$Z$ ← Partition function $Z := \sum_{S \subseteq V} \exp(F_\theta(S;V))$

**DeepSet for Permutation Invariance:**




Zaheer Kottur, Ravanbakhsh, Poczos, Salakhutdinov and Smola. Deep sets. NIPS 2017.

# Approximate MLE

Tencent AI Lab

$$p_\theta(S|V) = \frac{\exp(F_\theta(S;V))}{Z}$$

$Z$ ← Partition function $Z \coloneqq \sum_{S \subseteq V} \exp(F_\theta(S;V))$

**Training Discrete EBMs:**

Contrastive Divergence ⇒ Hard to converge

Score Matching ⇒ NonDifferentiable

Ratio Matching ⇒ Unstable

**Separate training and inference procedure** 😤

# Approximate MLE

$$p_\theta(S|V) = \frac{\exp(F_\theta(S;V))}{Z}$$

← Partition function $Z \coloneqq \sum_{S \subseteq V} \exp(F_\theta(S;V))$

**Marginal-based Loss:**

$\psi \in [0,1]^{|V|}$: odds that $s \in V$ shall be selected in the OS $S^*$

$$\psi^* = \underset{\psi}{\operatorname{argmax}} \, D(q(S;\psi) \| p_\theta(S))$$

$$L(\theta;\psi^*) = \sum_{i=1}^{N} [-\sum_{j \in S_i^*} \log \psi_j^* - \sum_{j \in V_i \backslash S_i^*} \log(1-\psi_j^*)]$$

**Cohesive training and inference procedure** 😀

Domke J. Learning graphical model parameters with approximate marginal inference. PAMI 2013.

# Approximate MLE

Tencent AI Lab

$$p_\theta(S|V) = \frac{\exp(F_\theta(S;V))}{Z}$$

← Partition function $Z := \sum_{S \subseteq V} \exp(F_\theta(S;V))$

**Marginal-based Loss:**

$\psi \in [0,1]^{|V|}$: odds that $s \in V$ shall be selected in the OS $S^*$

$$\psi^* = \underset{\psi}{\operatorname{argmax}}\, D(q(S;\psi) \| p_\theta(S))$$

$$L(\theta;\psi^*) = \sum_{i=1}^{N} [-\sum_{j \in S_i^*} \log \psi_j^* - \sum_{j \in V_i \setminus S_i^*} \log(1-\psi_j^*)]$$

Require $\psi^*$ is differentiable w.r.t. $\theta$ 🤔

# Differentiable MFVI

Variational distribution $q(S;\psi) = \prod_{i\in S}\psi_i \prod_{j\notin S}(1-\psi_j)\,, \psi \in [0,1]^{|V|}$

$$\min_{\psi} \mathbb{KL}(q(S;\psi)||p_\theta(S))$$

$$\Leftrightarrow \max_{\psi} f_{mt}^{F_\theta}(\psi) + \mathbb{H}\big(q(S;\psi)\big) =: \mathrm{ELBO}$$

multilinear extension $f_{mt}^{F_\theta}(\psi) \coloneqq \sum_{S\subseteq V} F_\theta(S) \prod_{i\in S}\psi_i \prod_{j\notin S}(1-\psi_j)$

# Differentiable MFVI

Tencent AI Lab

Variational distribution $q(S;\psi) = \prod_{i \in S} \psi_i \prod_{j \notin S} (1-\psi_j), \psi \in [0,1]^{|V|}$

$$\min_{\psi} \mathbb{KL}(q(S;\psi) \| p_\theta(S))$$

$$\Leftrightarrow \max_{\psi} f_{mt}^{F_\theta}(\psi) + \mathbb{H}\big(q(S;\psi)\big) =: \text{ELBO}$$

multilinear extension $f_{mt}^{F_\theta}(\psi) := \sum_{S \subseteq V} F_\theta(S) \prod_{i \in S} \psi_i \prod_{j \notin S} (1-\psi_j)$

**RNN-like fixed-point iteration:**

$$\left.\begin{aligned} &\psi^{(0)} \leftarrow \text{Initialize in } [0,1]^{|V|} \\ &\psi_\theta^{(k)} \leftarrow \left(1 + \exp\left(-\nabla_{\psi^{(k-1)}} f_{mt}^{F_\theta}\left(\psi^{(k-1)}\right)\right)\right)^{-1} \\ &\psi_\theta^{*} \leftarrow \psi_\theta^{(K)} \end{aligned}\right\} \text{MFVI}(\psi^{(0)}, V, K)$$

$\psi_\theta^{*} = \text{MFVI}\big(\psi^{(0)}, V, K\big)$ is **differentiable** w.r.t. $\theta$ 😌

# Differentiable MFVI

$$L(\theta;\psi^*) = \sum_{i=1}^{N}[-\sum_{j\in S_i^*}\log\psi_j^* - \sum_{j\in V_i\setminus S_i^*}\log(1-\psi_j^*)]$$

**Algorithm** $\mathrm{DiffMF}(V,S^*)$:

Initialize variational parameter $\psi$

$$\psi^{(0)} \leftarrow 0.5 * \mathbf{1}$$

Compute the variational marginals

$$\psi^* \leftarrow \mathrm{MFVI}(\psi^{(0)}, V, K)$$

Update parameter $\theta$

$$\theta \leftarrow \theta - \eta\nabla_\theta L(\theta;\psi^*)$$

# Differentiable MFVI

Tencent AI Lab

$$L(\theta;\psi^*) = \sum_{i=1}^{N}[-\sum_{j\in S_i^*}\log\psi_j^* - \sum_{j\in V_i\backslash S_i^*}\log(1-\psi_j^*)]$$

**Training:**

Initialize variational parameter $\psi$

$$\psi^{(0)} \leftarrow 0.5 * \mathbf{1}$$

Compute the variational marginals

$$\psi^* \leftarrow \mathrm{MFVI}(\psi^{(0)}, V, K)$$

Update parameter $\theta$

$$\theta \leftarrow \theta - \eta\nabla_\theta L(\theta;\psi^*)$$

**Inference:**

$$S = \mathrm{topN}(\psi^*)$$

$$\psi^* = \mathrm{MFVI}(\psi^{(0)}, V, K)$$

# Amortizing MFVI Open problems of DiffMF

Tencent AI Lab

- Expensive computation complexity

$$\psi_\theta^{(k)} \leftarrow \left(1 + \exp\left(-\nabla_{\psi^{(k-1)}} f_{mt}^{F_\theta}\left(\psi^{(k-1)}\right)\right)\right)^{-1}$$

expensive sampling loop per data point

- Discard interaction pattern

$$q(S;\psi) = \prod_{i\in S} \psi_i \prod_{j\notin S}(1-\psi_j)\,, \psi \in [0,1]^{|V|}$$

Independent assumption

# Amortizing MFVI Reduce computational complexity

$$\psi_\theta^{(k)} \leftarrow \left(1 + \exp\left(-\nabla_{\psi^{(k-1)}} f_{mt}^{F_\theta}\left(\psi^{(k-1)}\right)\right)\right)^{-1}$$

expensive sampling loop per data point

Approximate with neural networks:

neural network $q_\phi: \mathbb{R}^{|V|} \rightarrow \mathbb{R}^{|V|}$

$$L = \mathbb{KL}(q_\phi(S;\psi) \| p_\theta(S))$$
$$= f_{mt}^{F_\theta}(\psi) + \mathbb{H}\left(q_\phi(S;\psi)\right) + const$$

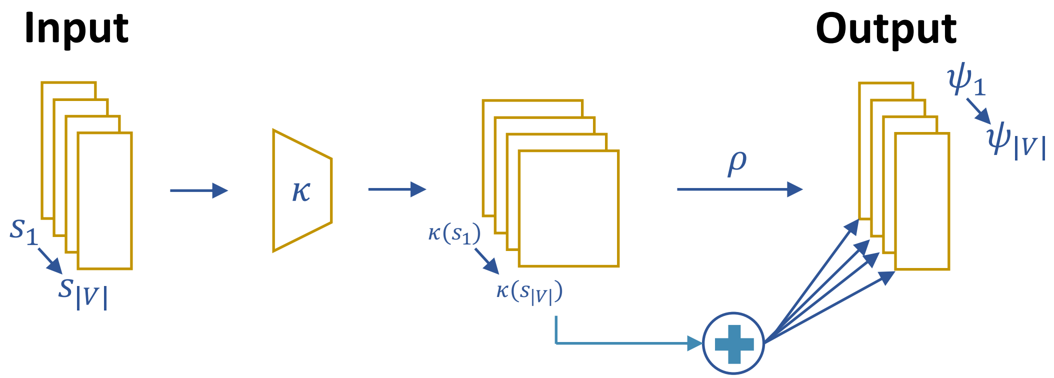
$q_\phi(S;\psi)$ should satisfy **equivariant**

# Amortizing MFVI Reduce computational complexity

$$\psi_\theta^{(k)} \leftarrow \left(1 + \exp\left(-\nabla_{\psi^{(k-1)}} f_{mt}^{F_\theta}\left(\psi^{(k-1)}\right)\right)\right)^{-1}$$

expensive sampling loop per data point

$\mathrm{EquiNet}(V;\phi)$ with permutation equivariance




Zaheer Kottur, Ravanbakhsh, Poczos, Salakhutdinov and Smola. Deep sets. NIPS 2017.

# Amortizing MFVI Induce interaction pattern

$$q(S;\psi) = \prod_{i\in S}\psi_i \prod_{j\notin S}(1-\psi_j)\,,\psi \in [0,1]^{|V|}$$

Independent assumption

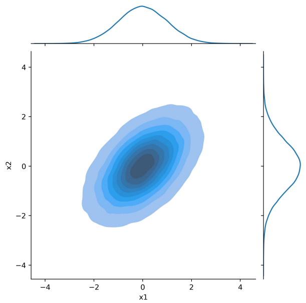
## Correlation-aware inference:

Copula density

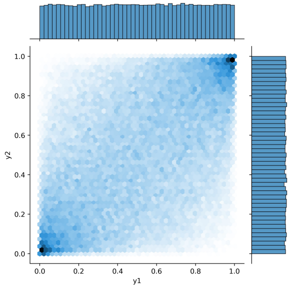
Marginal distribution

$$q(s_1, s_2, \dots, s_{|V|}) = \mathrm{c}\left(Q_1(s_1), Q_2(s_2), \dots, Q_{|V|}(s_{|V|})\right) \prod_{i=1}^{|V|} q_i(s_i)$$

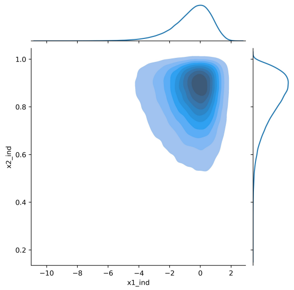
Cumulative distribution function



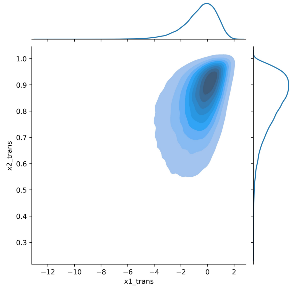
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim N(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix})$$



$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \sim c(Q_1(x_1), Q_2(x_2))$$



Copula



$${x_1}_{trans} \sim Gumbel$$
$${x_2}_{trans} \sim Beta$$

# Amortizing MFVI Induce interaction pattern

Tencent AI Lab

$$q(s_1, s_2, \dots, s_{|V|}) = \mathrm{c}\left(Q_1(s_1), Q_2(s_2), \dots, Q_{|V|}(s_{|V|})\right) \prod_{i=1}^{|V|} q_i(s_i)$$

Apply Gaussian copula here

Induce Gaussian copula:

Sample an auxiliary noise

$$g \sim N(0, \Sigma)$$ → Covariance matrix , parameterized by neural network

Apply element-wise Gaussian CDF

$$u = \Phi_{diag(\Sigma)}(g)$$

Obtain binary sample

$$s = \mathbb{I}(\psi \leq u)$$ → Original logits of Bernoulli

# Amortizing MFVI

$$\phi^* = \underset{\phi}{\operatorname{argmax}} f_{mt}^{F_\theta}(\psi) + \mathbb{H}\left(q_\phi(S;\psi)\right) := \text{ELBO}$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} [-\sum_{j\in S_i^*} \log \psi_j^* - \sum_{j\in V_i \backslash S_i^*} \log(1-\psi_j^*)]$$

$\psi^* = \text{MFVI}(\psi^{(0)}, V, K)$

**Algorithm** $\text{EquiVSet}(V, S^*)$:

Update parameter $\phi$

$$\phi \leftarrow \phi + \eta \nabla_\phi \text{ELBO}(\phi)$$

→ Optimize $\phi$

Initialize variational parameter

$$\psi^{(0)} \leftarrow \text{EquiNet}(V;\phi)$$

One step fixed point iteration

$$\psi^* \leftarrow \text{MFVI}(\psi^{(0)}, V, K=1)$$

Mean-field Iteration

Update parameter $\theta$

$$\theta \leftarrow \theta - \eta \nabla_\theta L(\theta;\psi^*)$$

→ Optimize $\theta$

# Application: Product Recommender

Tencent AI Lab

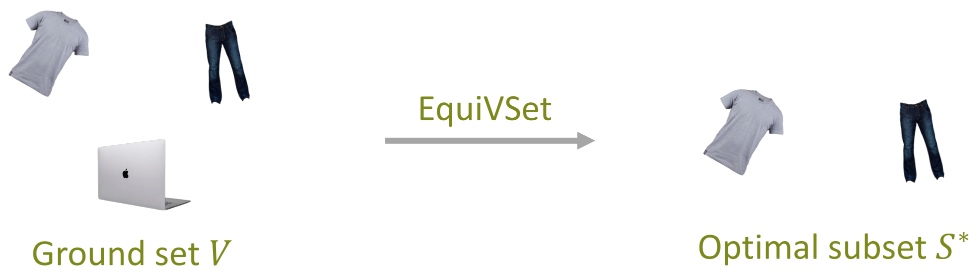



Table 2: Product recommendation results in the MJC metric on the Amazon dataset.

| Categories | Random | PGM | DeepSet (NoSetFn) | DiffMF (ours) | EquiVSet$_{\text{ind}}$ (ours) | EquiVSet$_{\text{copula}}$ (ours) |
|---|---|---|---|---|---|---|
| Toys | 0.0832 | 0.4414 ± 0.0036 | 0.4287 ± 0.0047 | 0.6147 ± 0.0102 | 0.6491 ± 0.0152 | **0.6762 ± 0.0221** |
| Furniture | 0.0651 | 0.1746 ± 0.0069 | 0.1758 ± 0.0072 | 0.1744 ± 0.0121 | **0.1775 ± 0.0108** | 0.1724 ± 0.0091 |
| Gear | 0.0771 | 0.4712 ± 0.0037 | 0.3806 ± 0.0019 | 0.5622 ± 0.0171 | 0.6103 ± 0.0193 | **0.6973 ± 0.0119** |
| Carseats | 0.0659 | **0.2330 ± 0.0115** | 0.2121 ± 0.0096 | 0.2229 ± 0.0104 | 0.2141 ± 0.0073 | 0.2149 ± 0.0123 |
| Bath | 0.0763 | 0.5638 ± 0.0077 | 0.4241 ± 0.0058 | 0.6901 ± 0.0061 | 0.6457 ± 0.0200 | **0.7567 ± 0.0095** |
| Health | 0.0758 | 0.4493 ± 0.0024 | 0.4481 ± 0.0041 | 0.5650 ± 0.0092 | 0.6315 ± 0.0153 | **0.7003 ± 0.0159** |
| Diaper | 0.0839 | 0.5802 ± 0.0092 | 0.4572 ± 0.0050 | 0.7011 ± 0.0112 | 0.7344 ± 0.0199 | **0.8275 ± 0.0136** |
| Bedding | 0.0791 | 0.4799 ± 0.0061 | 0.4824 ± 0.0081 | 0.6408 ± 0.0093 | 0.6287 ± 0.0195 | **0.7688 ± 0.0121** |
| Safety | 0.0648 | 0.2495 ± 0.0060 | 0.2211 ± 0.0044 | 0.2007 ± 0.0527 | 0.2250 ± 0.0287 | **0.2524 ± 0.0285** |
| Feeding | 0.0925 | 0.5596 ± 0.0081 | 0.4295 ± 0.0021 | 0.7496 ± 0.0114 | 0.6955 ± 0.0063 | **0.8101 ± 0.0074** |
| Apparel | 0.0918 | 0.5333 ± 0.0050 | 0.5074 ± 0.0036 | 0.6708 ± 0.0225 | 0.6465 ± 0.0150 | **0.7521 ± 0.0114** |
| Media | 0.0944 | 0.4406 ± 0.0092 | 0.4241 ± 0.0105 | 0.5145 ± 0.0105 | 0.5506 ± 0.0072 | **0.5694 ± 0.0105** |

Metric: $\mathrm{MJC} := \frac{1}{|\mathcal{D}_t|} \sum_{(V,S^*)\in\mathcal{D}_t} \frac{|S^* \cap S|}{|S^* \cup S|}$

# Application: Anomaly Detection



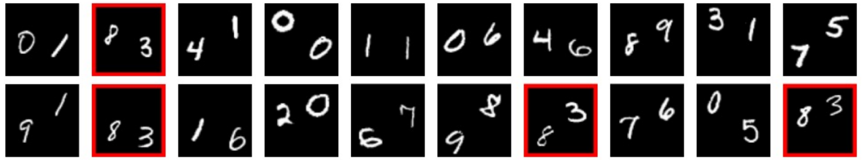
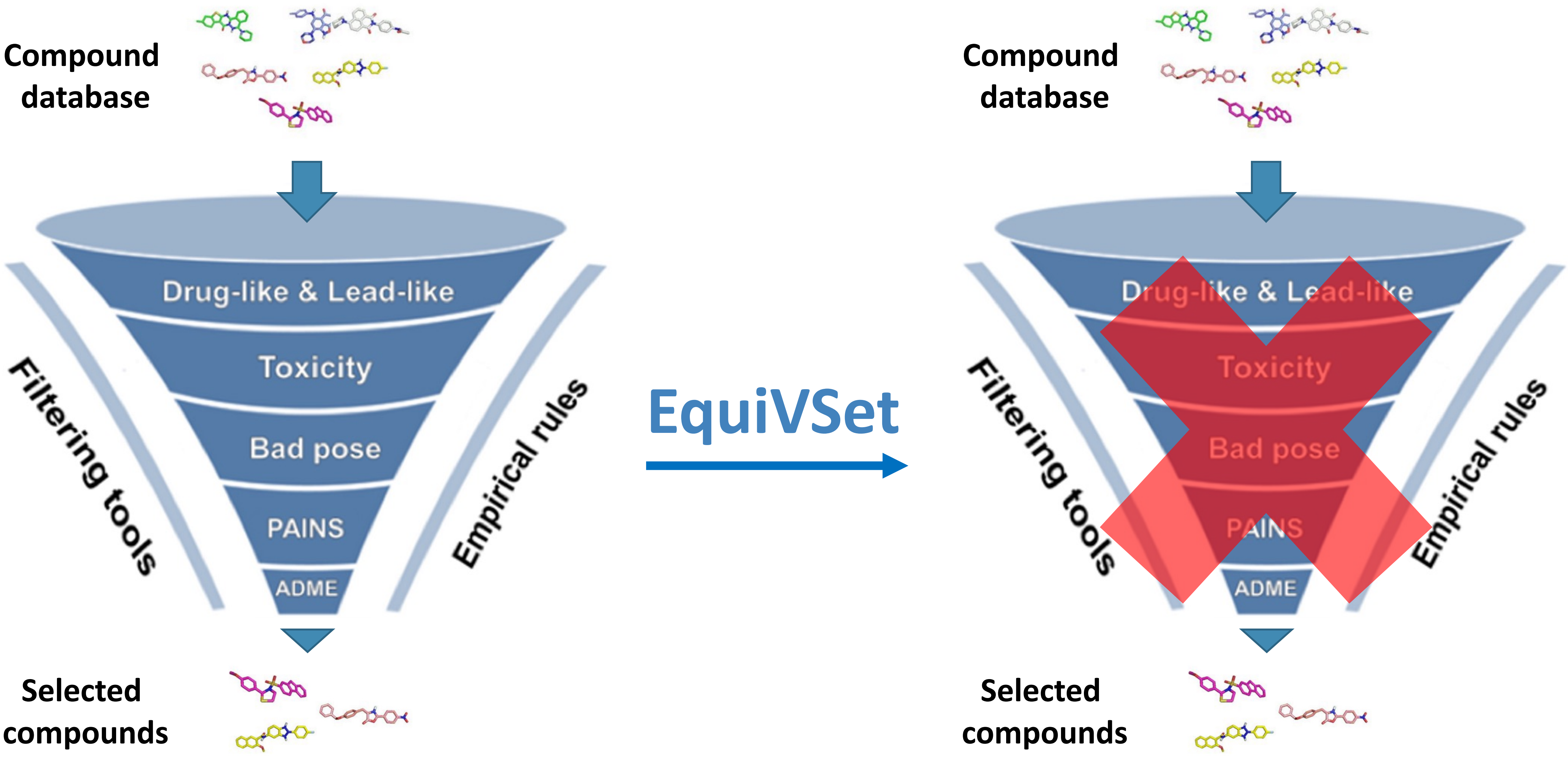



Table 3: Set anomaly detection results in the MJC metric.

| Method | Double MNIST | CelebA |
|---|---|---|
| Random | 0.0816 | 0.2187 |
| PGM | $0.3031 \pm 0.0118$ | $0.4812 \pm 0.0064$ |
| DeepSet (NoSetFn) | $0.1108 \pm 0.0031$ | $0.3915 \pm 0.0133$ |
| `DiffMF` (ours) | $\mathbf{0.6064} \pm \mathbf{0.0133}$ | $0.5455 \pm 0.0079$ |
| `EquiVSet`$_{\text{ind}}$ (ours) | $0.4054 \pm 0.0122$ | $0.5310 \pm 0.0123$ |
| `EquiVSet`$_{\text{copula}}$ (ours) | $0.5878 \pm 0.0068$ | $\mathbf{0.5549} \pm \mathbf{0.0053}$ |

# Application: Compound Selection




Lionta et al. Structure-based virtual screening for drug discovery: principles, applications and recent advances. Current Topics in Medicinal Chemistry, 2014.

# Application: Compound Selection

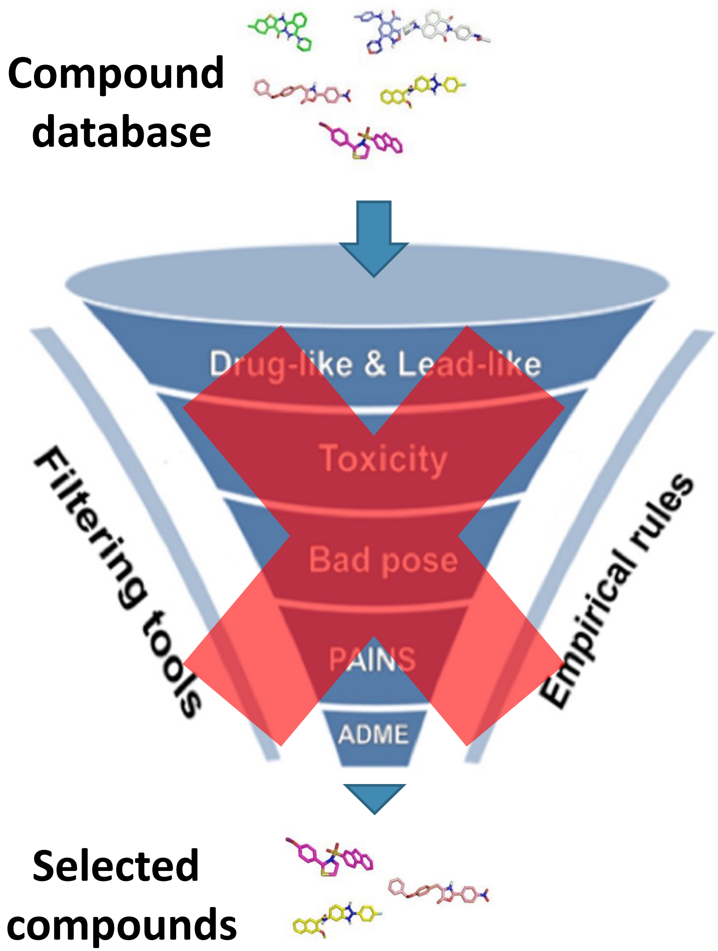



Table 4: Compound selection results in the MJC metric.

| Method | PDBBind | BindingDB |
|---|---|---|
| Random | 0.0725 | 0.0267 |
| PGM | $0.3499 \pm 0.0087$ | $0.1760 \pm 0.0055$ |
| DeepSet (NoSetFn) | $0.3189 \pm 0.0034$ | $0.1615 \pm 0.0074$ |
| DiffMF (ours) | $0.3534 \pm 0.0143$ | $0.1894 \pm 0.0021$ |
| $\texttt{EquiVSet}_{\text{ind}}$ (ours) | $\mathbf{0.3553 \pm 0.0049}$ | $\mathbf{0.1904 \pm 0.0034}$ |
| $\texttt{EquiVSet}_{\text{copula}}$ (ours) | $0.3536 \pm 0.0083$ | $0.1875 \pm 0.0032$ |

Tencent AI Lab

# Thank you!