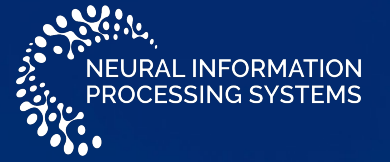




Tencent  
AI Lab

Imperial College  
London



NEURAL INFORMATION  
PROCESSING SYSTEMS

(Selected as Oral)

# Learning Neural Set Functions Under the Optimal Subset Oracle

Zijing Ou<sup>1</sup>, Tingyang Xu<sup>2</sup>, Qinliang Su<sup>3</sup>, Yingzhen Li<sup>1</sup>, Peilin Zhao<sup>2</sup>, Yatao Bian<sup>2</sup>

<sup>1</sup>Imperial College London

<sup>2</sup>Tencent AI Lab

<sup>3</sup>Sun Yat-sen University

# Background of Set Function Learning



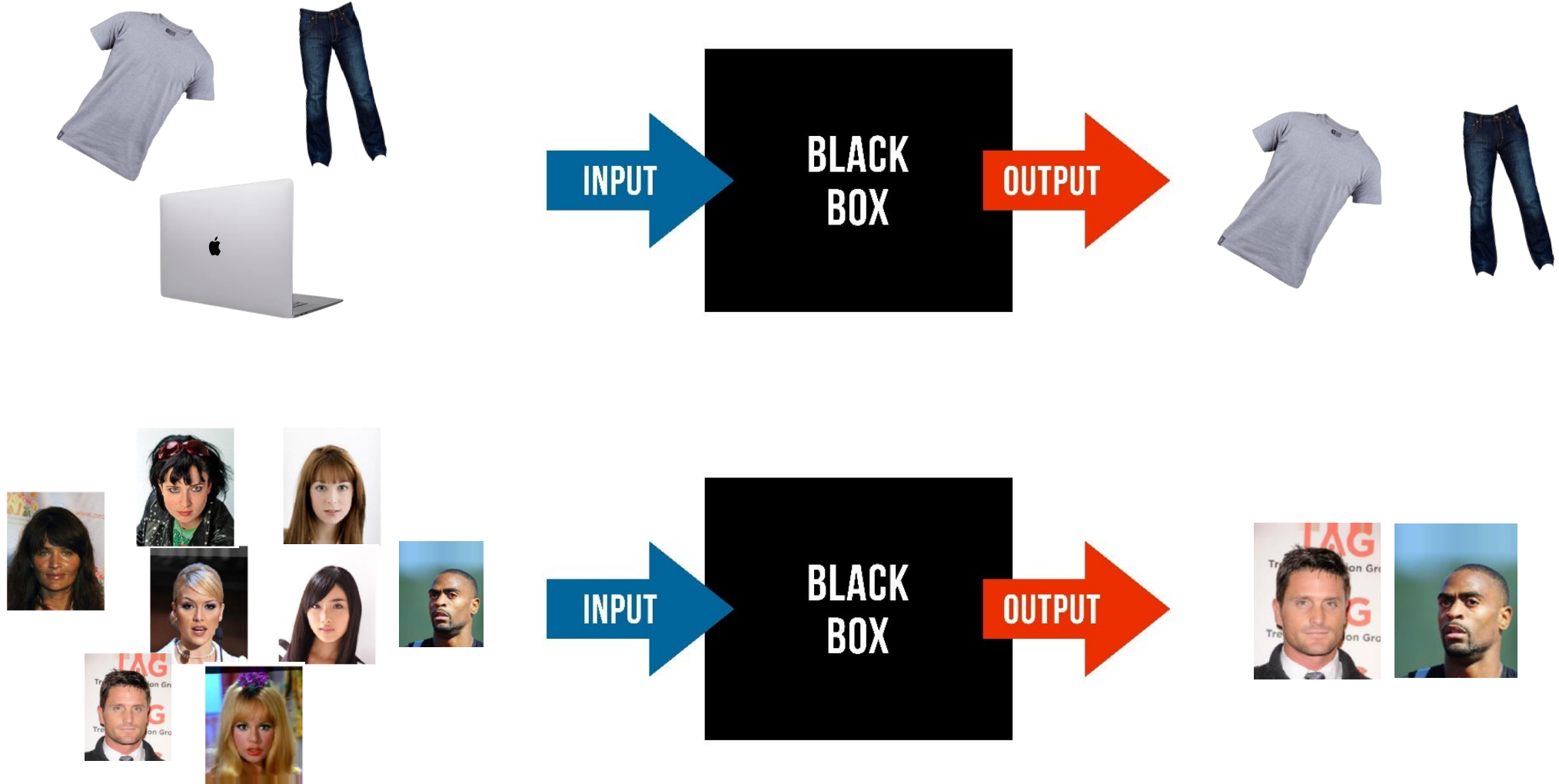
Product Recommendation



Anomaly Detection



# Background of Set Function Learning



# Background of Set Function Learning



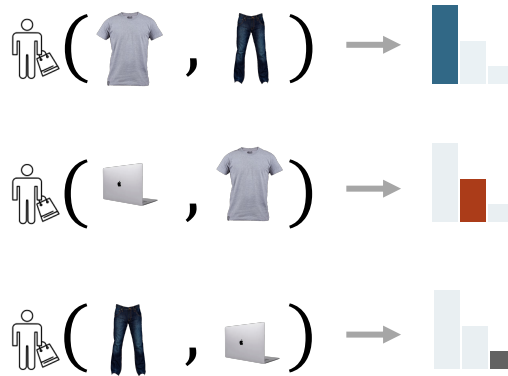
Database



Ground set  $V$



Customer



Utility function  
 $F_{\theta^*}(\cdot; V): 2^V \rightarrow \mathbb{R}$



Shopping cart



Optimal subset  $S^*$

**Data Generation Process:**

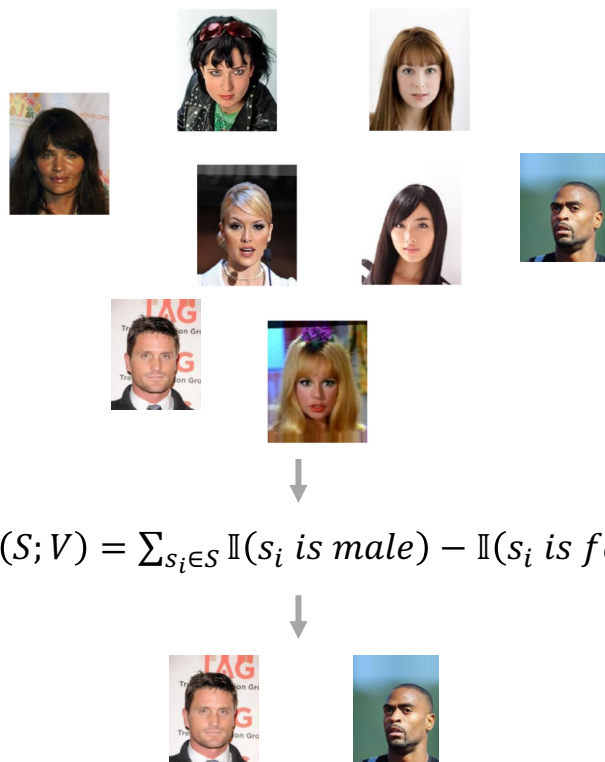
$$S^* = \operatorname{argmax}_{S \in 2^V} F_{\theta^*}(S; V)$$

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

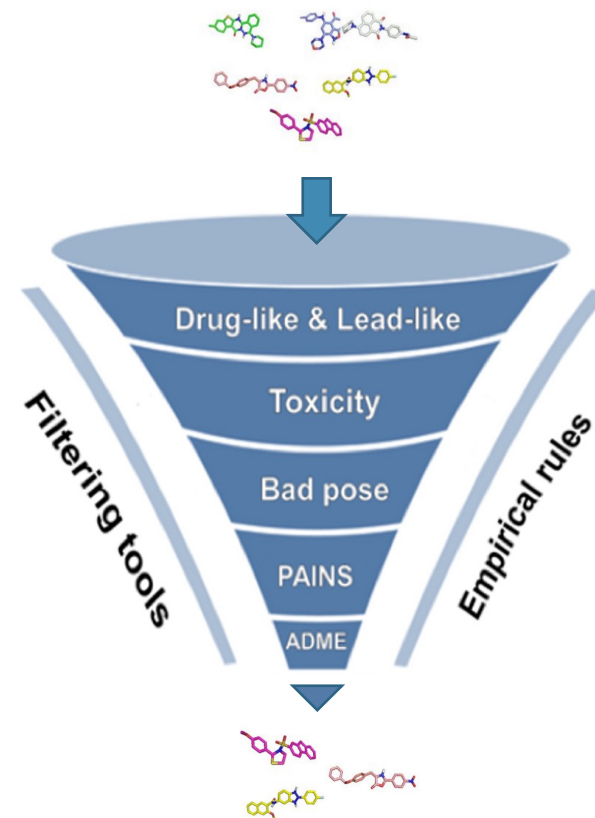
# Background of Set Function Learning

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

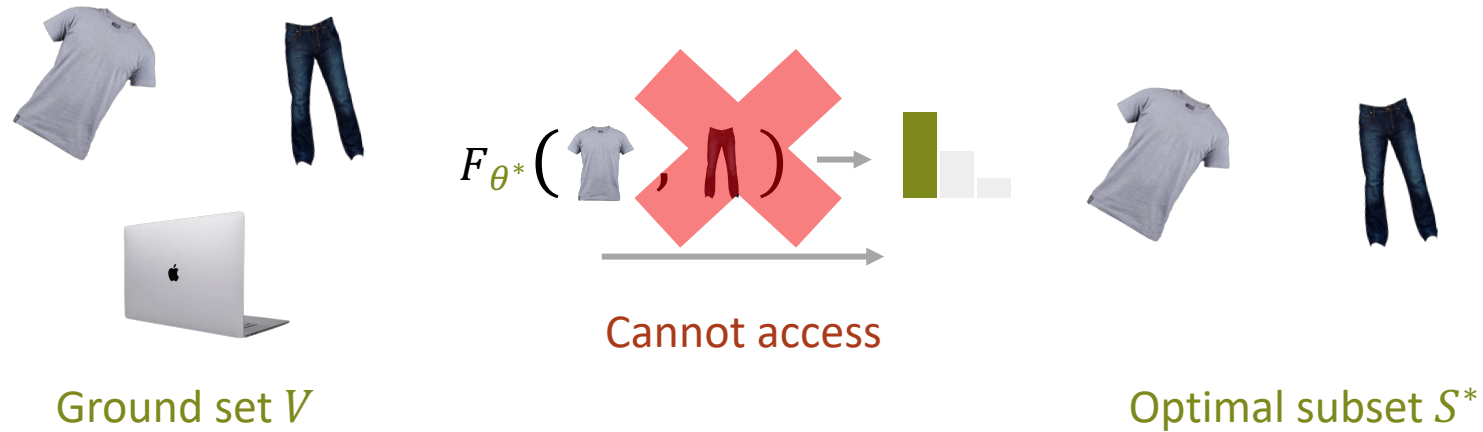
## Set anomaly detection



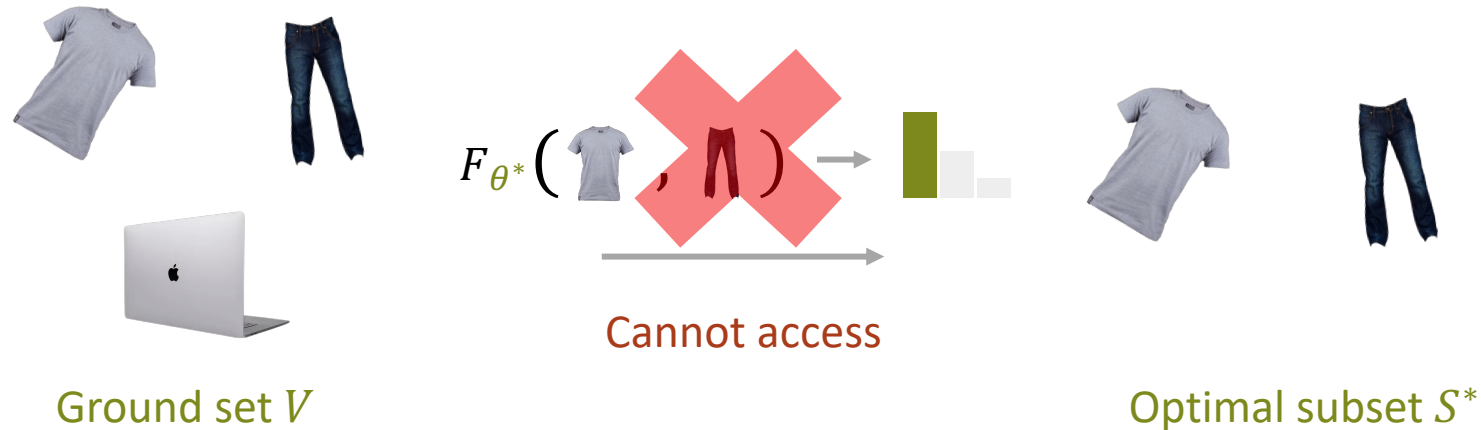
## Compound selection



**Setting:** Training data is given in form of  $\{(V_i, S_i^*)\}_{i=1}^N$



**Setting:** Training data is given in form of  $\{(V_i, S_i^*)\}_{i=1}^N$



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

$$S^* = \operatorname{argmax}_{S \in 2^V} F_{\theta}(S; V), \forall (V, S^*) \in \{(V_i, S_i^*)\}_{i=1}^N$$

**Maximum Likelihood:**

$$\begin{aligned} & \text{Empirical distribution} \\ & \downarrow \\ & \operatorname{argmax}_{\theta} \mathbb{E}_{\mathbb{P}(S^*, V)} [\log p_{\theta}(S^* | V)] \\ & s. t. p_{\theta}(S | V) \propto F_{\theta}(S; V), \forall S \in 2^V \\ & \uparrow \\ & \text{Monotonically grows with the utility function} \end{aligned}$$



**Maximum Likelihood:**

$$\begin{aligned} & \text{Empirical distribution} \\ & \downarrow \\ & \operatorname{argmax}_{\theta} \mathbb{E}_{\mathbb{P}(S^*, V)} [\log p_{\theta}(S^* | V)] \\ & s. t. p_{\theta}(S | V) \propto F_{\theta}(S; V), \forall S \in 2^V \\ & \uparrow \\ & \text{Monotonically grows with the utility function} \end{aligned}$$

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

## Maximum Likelihood:

$$\begin{aligned} & \operatorname{argmax}_{\theta} \mathbb{E}_{\mathbb{P}(S^*, V)} [\log p_{\theta}(S^* | V)] \\ & \text{s.t. } p_{\theta}(S | V) \propto F_{\theta}(S; V), \forall S \in 2^V \end{aligned}$$

## Desiderata:

*Permutation invariance*

$$F_{\theta}(\text{👕}, \text{👖}) = F_{\theta}(\text{👖}, \text{👕})$$

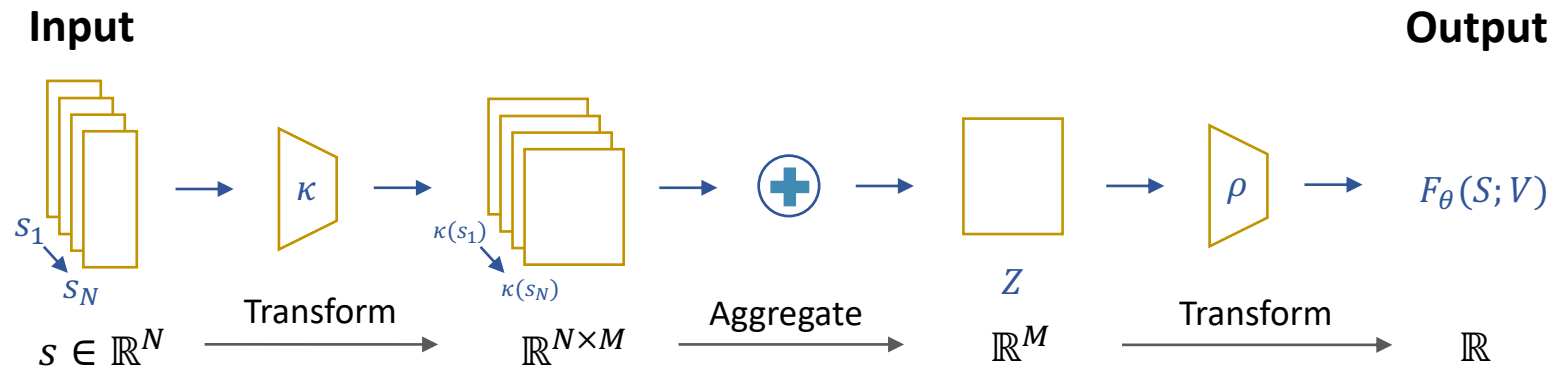
*Varying ground set*

$$F_{\theta}(\text{👕}; \text{👕} \text{ 👖} \text{ 📁}) \rightarrow \text{■} \qquad F_{\theta}(\text{👕}; \text{👕} \text{ 👖} \text{ 📁} \text{ 📱}) \rightarrow \text{■}$$

.....

$$p_{\theta}(S|V) = \frac{\exp(F_{\theta}(S; V))}{Z} \leftarrow \text{Partition function } Z := \sum_{S \subseteq V} \exp(F_{\theta}(S; V))$$

## DeepSet for Permutation Invariance & Varying Ground Set:



## Marginal-based Loss:

$\psi^*$  is differentiable w.r.t.  $\theta$   $q(S; \psi) = \prod_{i \in S} \psi_i \prod_{j \notin S} (1 - \psi_j), \psi \in [0, 1]^{|V|}$

$$\psi^{*|\theta} = \underset{\psi}{\operatorname{argmin}} \operatorname{KL}(q_\phi(S; \psi) \| p_\theta(S))$$

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

Training  $\theta$  by differentiating through  $\psi^*$  using cross entropy loss

## Marginal-based Loss:

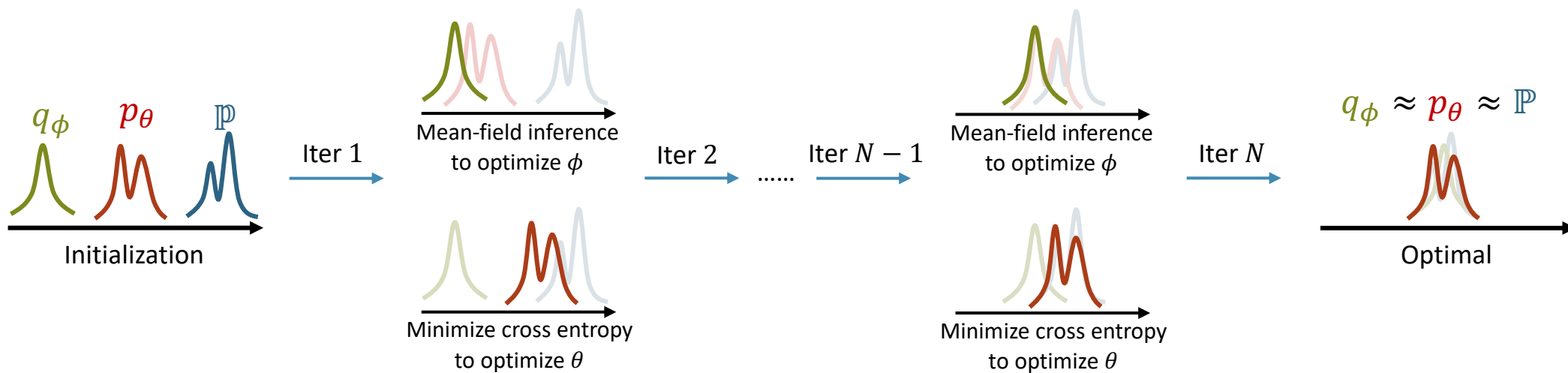
$\psi^*$  is differentiable w.r.t.  $\theta$

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

$$\psi^{*|\theta} = \underset{\psi}{\operatorname{argmin}} \operatorname{KL}(q_{\phi}(S; \psi) \| p_{\theta}(S))$$

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

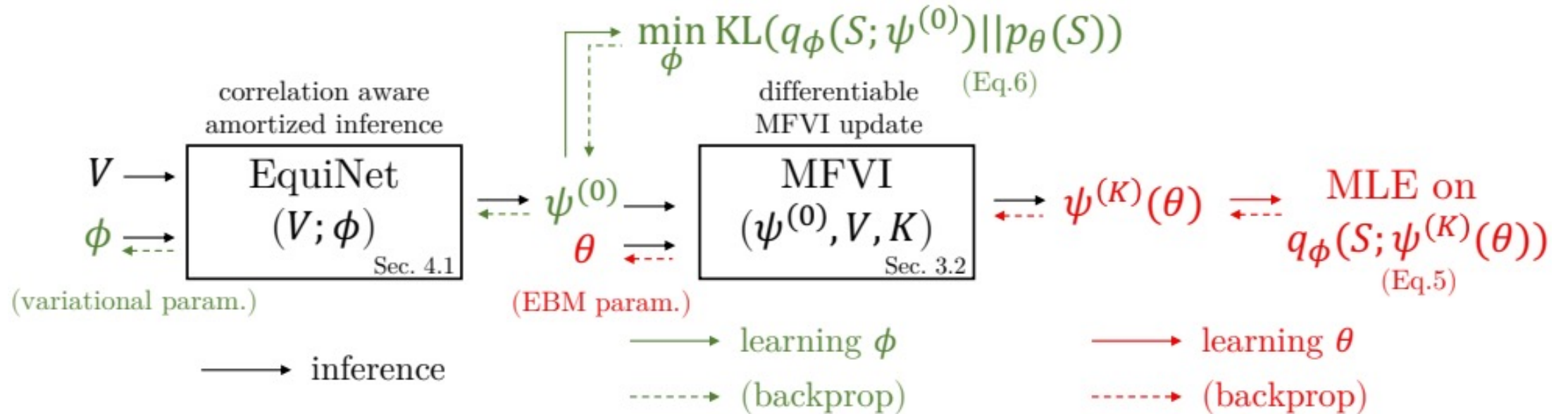
Training  $\theta$  by differentiating through  $\psi^*$  using cross entropy loss



$$\psi^{*|\theta} = \operatorname{argmin}_{\psi} \operatorname{KL}(q(S; \psi) \| p_{\theta}(S))$$

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

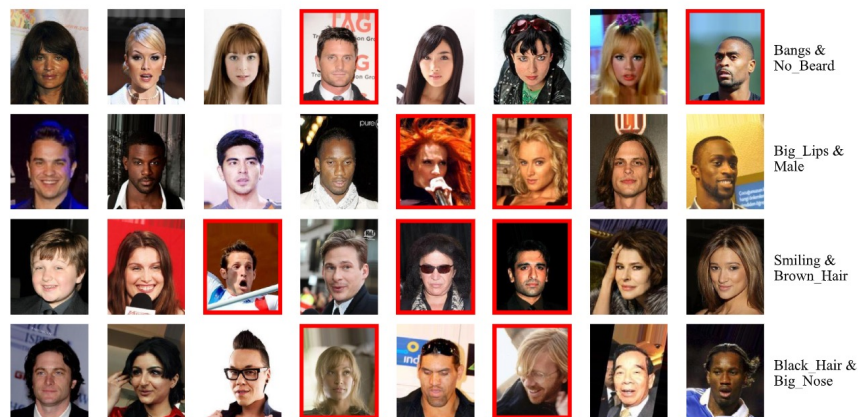
## Equivariant Variational inference for Set function learning (EquiVSet)



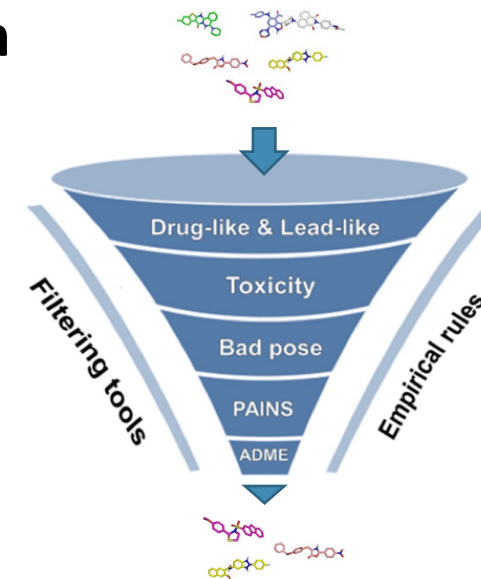
EquiVSet achieves the **BEST** results on 3 different tasks!



**Product Recommendation**



**Set Anomaly Detection**



**Compound Selection**

Baselines:

Random: random pick

PGM: probabilistic greedy models

DeepSet (NoSetFn): only the q net

Table 2: Product recommendation results on the Amazon dataset with different categories.

Categories	Random	PGM	DeepSet (NoSetFn)	DiffMF (ours)	EquiVSet <sub>ind</sub> (ours)	EquiVSet <sub>copula</sub> (ours)
Toys	0.083	0.441 ± 0.004	0.429 ± 0.005	0.610 ± 0.010	0.650 ± 0.015	<b>0.680 ± 0.020</b>
Furniture	0.065	0.175 ± 0.007	<b>0.176 ± 0.007</b>	0.170 ± 0.010	0.170 ± 0.011	0.172 ± 0.009
Gear	0.077	0.471 ± 0.004	0.381 ± 0.002	0.560 ± 0.020	0.610 ± 0.020	<b>0.700 ± 0.020</b>
Carseats	0.066	<b>0.230 ± 0.010</b>	0.210 ± 0.010	0.220 ± 0.010	0.214 ± 0.007	0.210 ± 0.010
Bath	0.076	0.564 ± 0.008	0.424 ± 0.006	0.690 ± 0.006	0.650 ± 0.020	<b>0.757 ± 0.009</b>
Health	0.076	0.449 ± 0.002	0.448 ± 0.004	0.565 ± 0.009	0.630 ± 0.020	<b>0.700 ± 0.020</b>
Diaper	0.084	0.580 ± 0.009	0.457 ± 0.005	0.700 ± 0.010	0.730 ± 0.020	<b>0.830 ± 0.010</b>
Bedding	0.079	0.480 ± 0.006	0.482 ± 0.008	0.641 ± 0.009	0.630 ± 0.020	<b>0.770 ± 0.010</b>
Safety	0.065	0.250 ± 0.006	0.221 ± 0.004	0.200 ± 0.050	0.230 ± 0.030	<b>0.250 ± 0.030</b>
Feeding	0.093	0.560 ± 0.008	0.430 ± 0.002	0.750 ± 0.010	0.696 ± 0.006	<b>0.810 ± 0.007</b>
Apparel	0.090	0.533 ± 0.005	0.507 ± 0.004	0.670 ± 0.020	0.650 ± 0.020	<b>0.750 ± 0.010</b>
Media	0.094	0.441 ± 0.009	0.420 ± 0.010	0.510 ± 0.010	0.551 ± 0.007	<b>0.570 ± 0.010</b>

Metric:

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

EquiVSet achieves improvements up to **33%** on average





Tencent  
AI Lab

Imperial College  
London



# Thank you!

Code is available at: <https://github.com/SubsetSelection/EquiVSet>

✓ Welcome to try it out in Colab:  Open in Colab