Domain Generalization with Small Data

International Journal of Computer Vision (IJCV) 2024

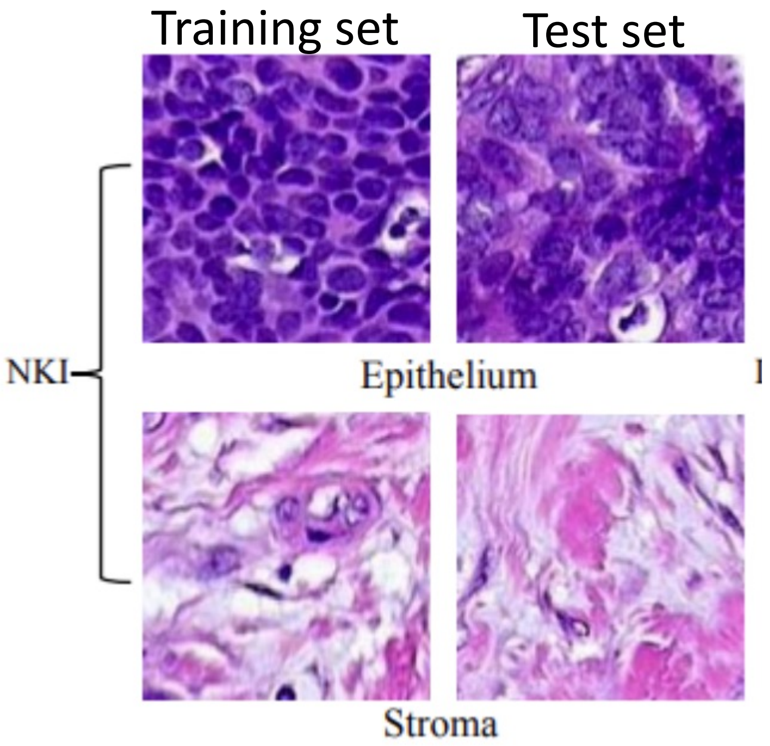
Kecheng Chen[1], Elena Gal[2], Hong Yan[1], Haoliang Li[1]

[1] Department of Electrical Engineering, City University of Hong Kong

[2] Department of Mathematics, University of Oxford

Motivation

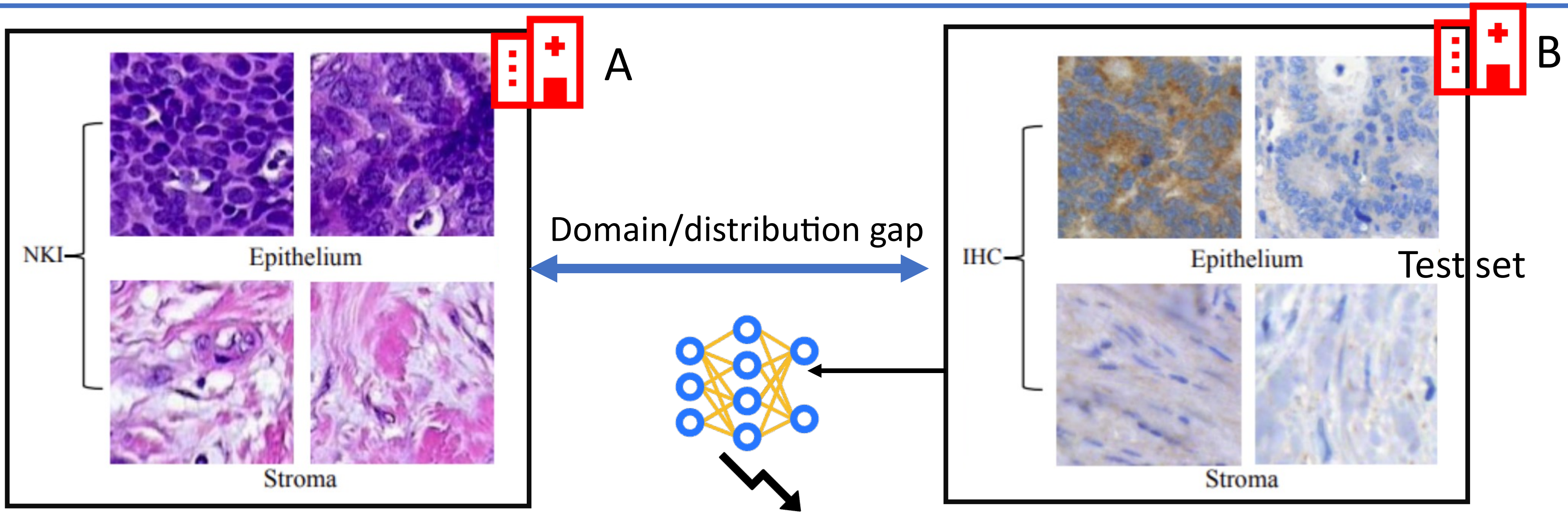
Common assumption : iid



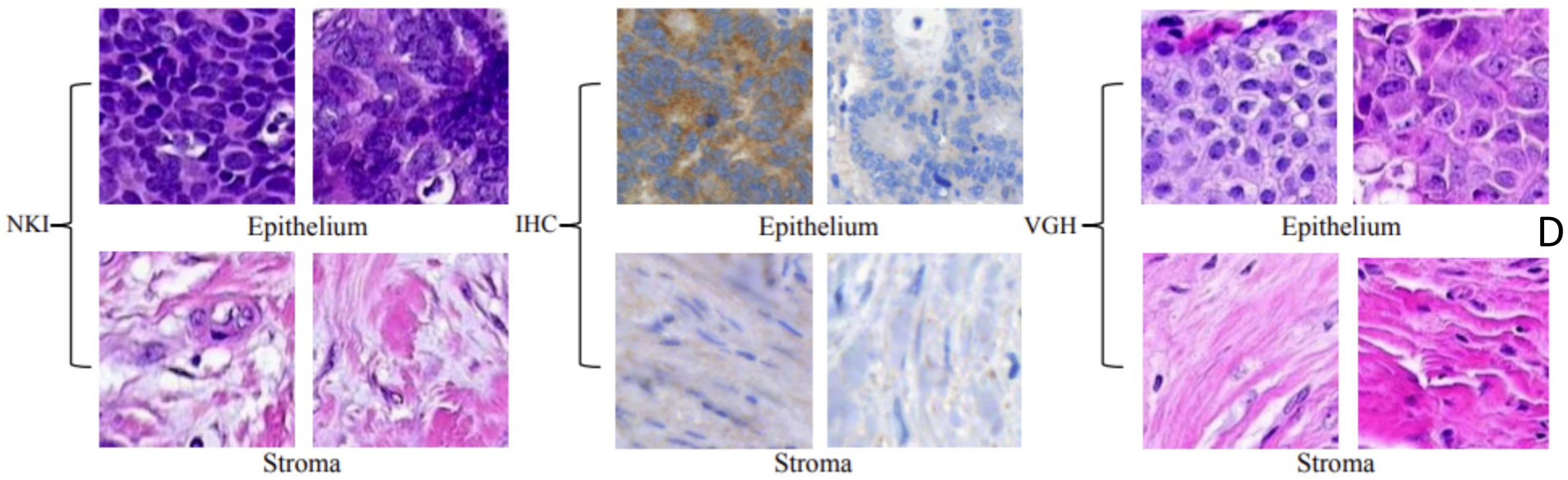
Out of distribution problem

Training set

Different colouring agents



What is Domain Generalization (DG) ?

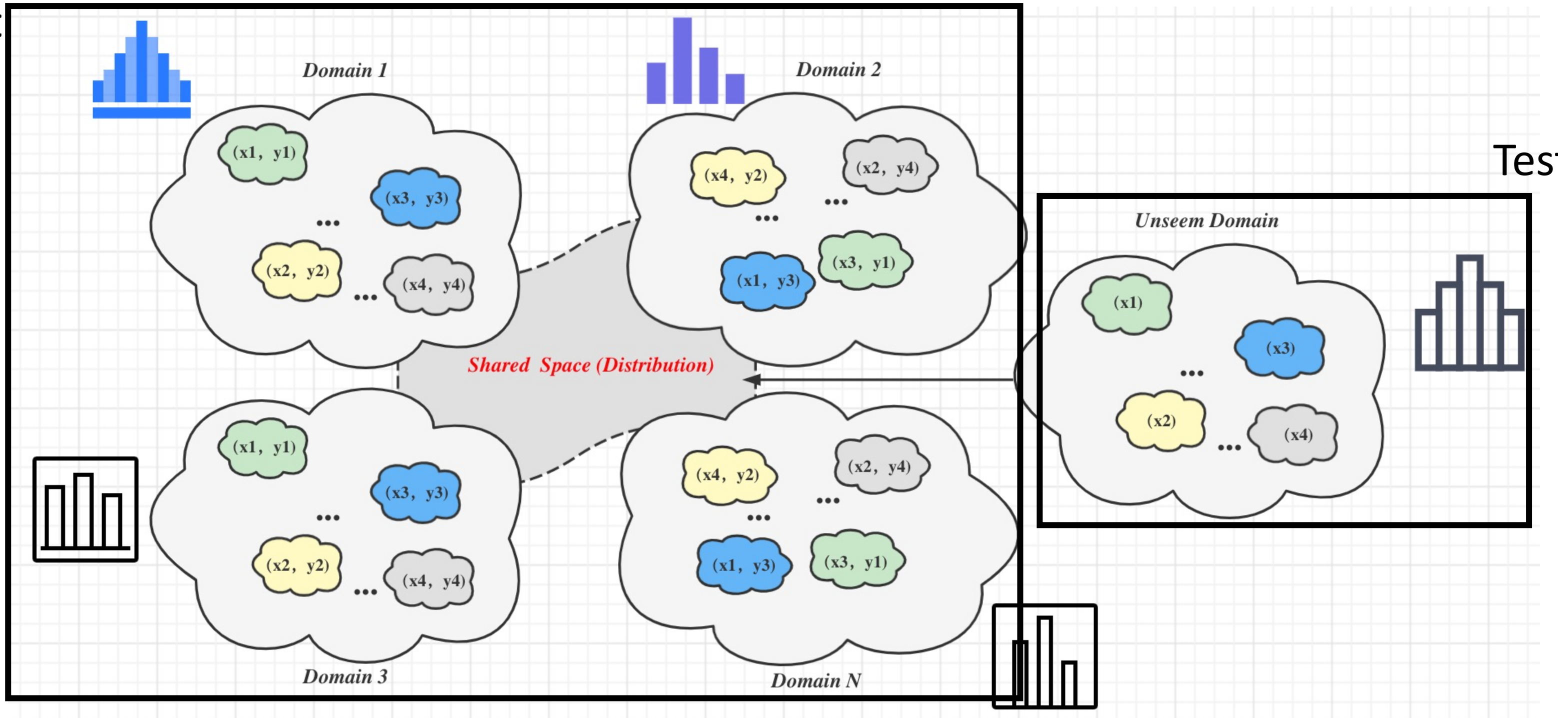


Different colouring agents

Training set

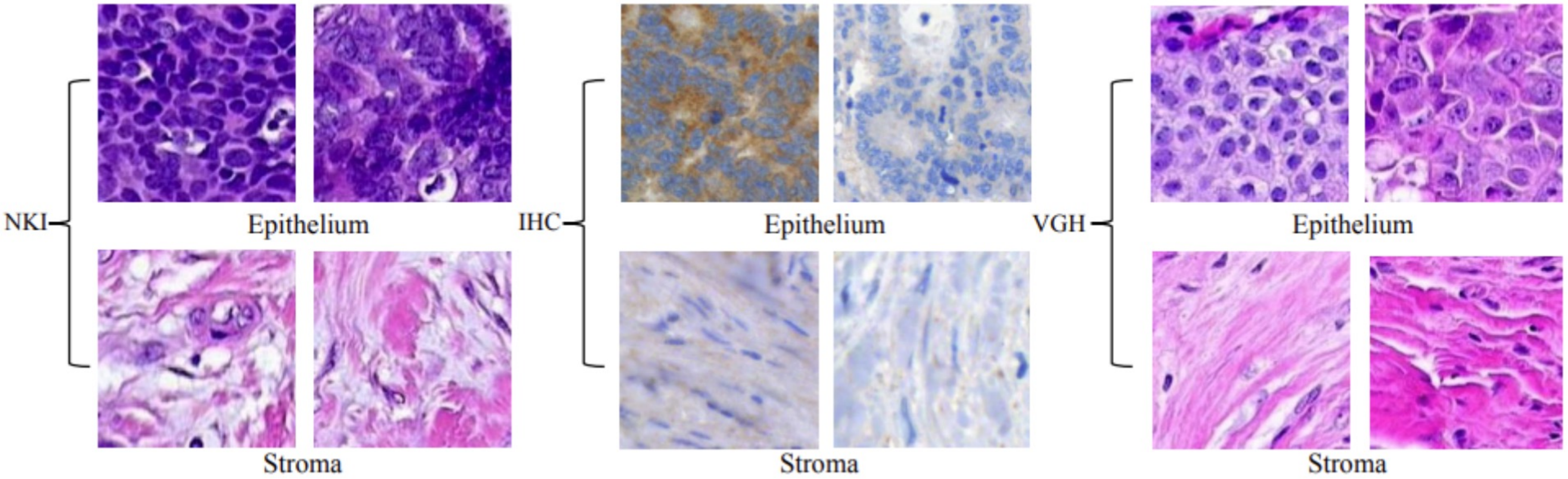
1. Obtain a more robust models

2. Annotating the data is expensive



Test set

What is DG in the context of small data?



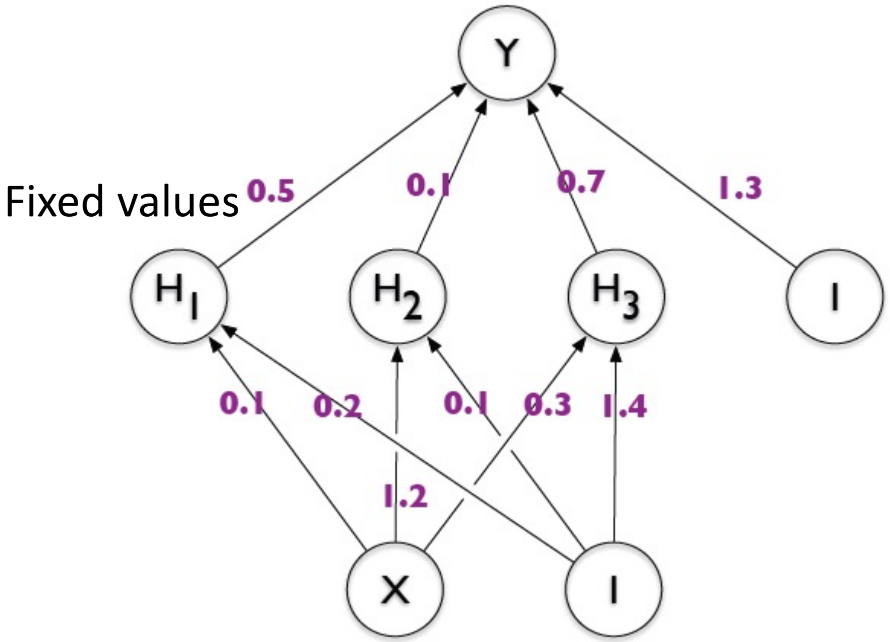
Healthcare Data due to potential privacy concerns or rare diseases

Chip design data due to IP protection

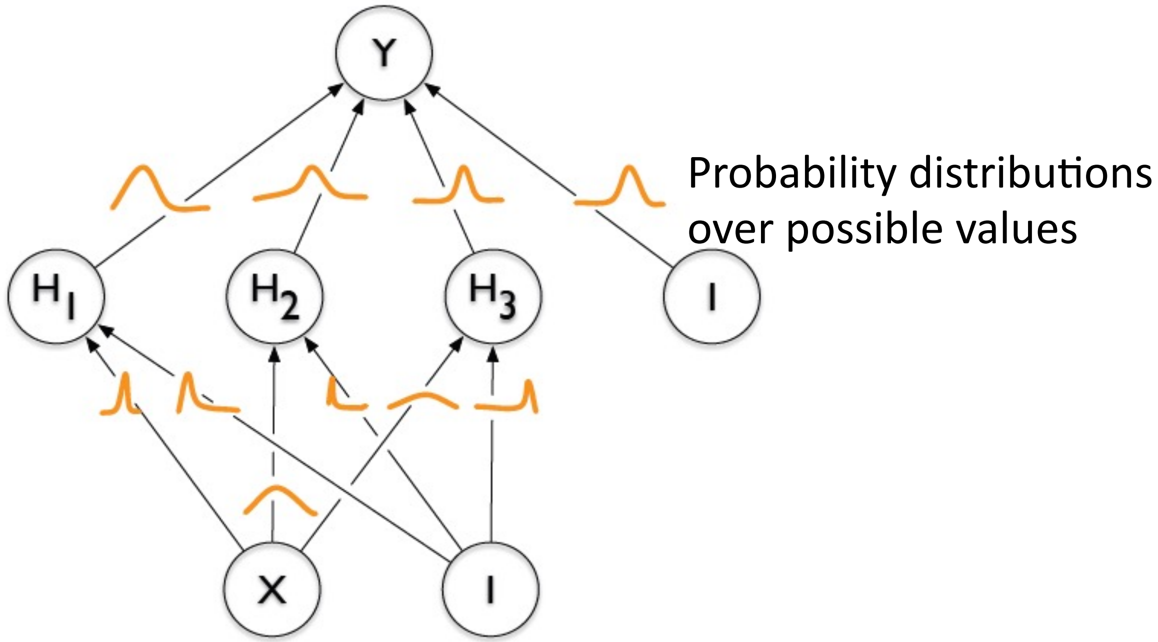
DG with Small Data

1️⃣ *Make the backbone more adaptive to small data scenarios*

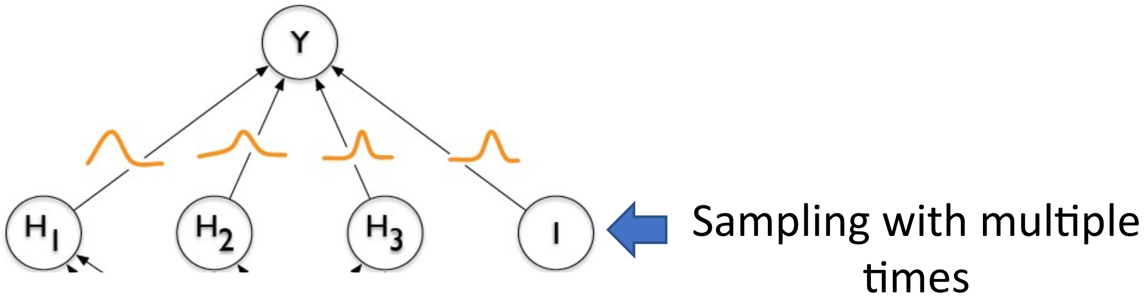
Deterministic Neural Networks (e.g., CNNs, FCNs)



Bayesian Neural Networks



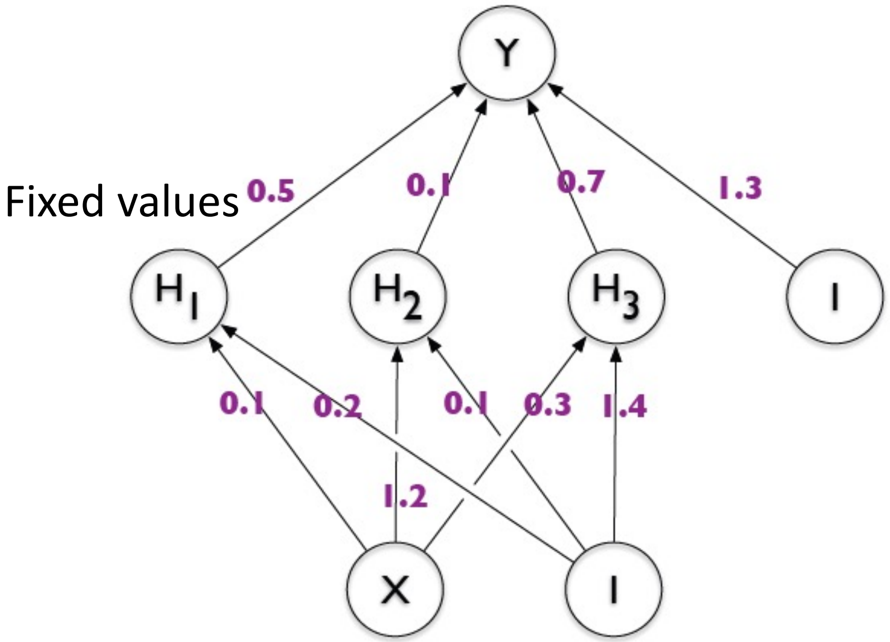
1. Richer representations and predictions from cheap model averaging.



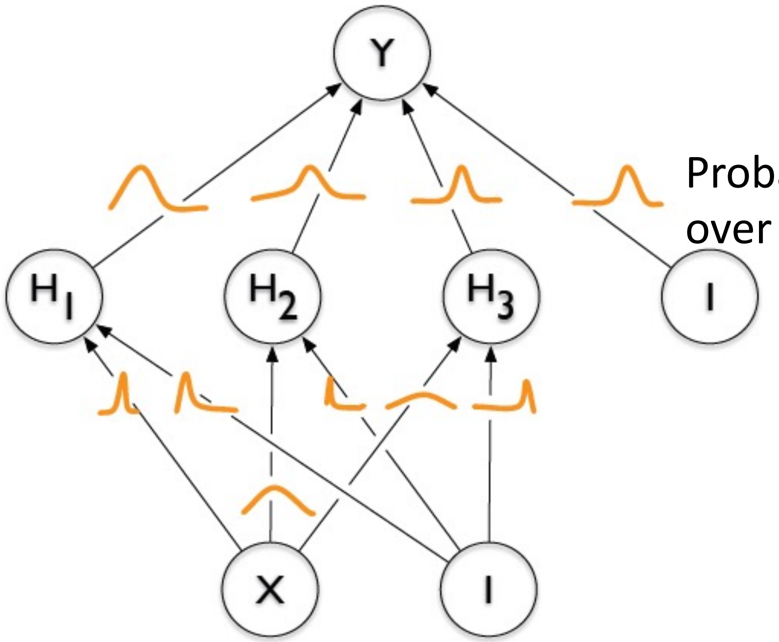
DG with Small Data

1️⃣ *Make the backbone more adaptive to small data scenarios*

Deterministic Neural Networks (e.g., CNNs, FCNs)



Bayesian Neural Networks



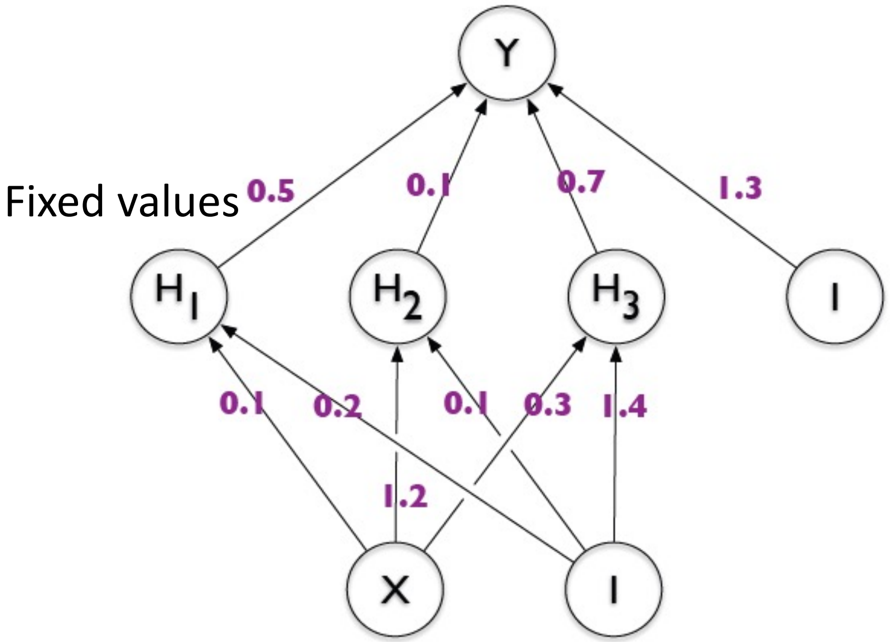
2. Model parameters can be regularized by a prior distribution with less overfitting risk in the context of small data scenarios

$$
\begin{aligned}
\theta^\star &= \arg\min_\theta \mathrm{KL}[q(\mathbf{w}|\theta)||P(\mathbf{w}|\mathcal{D})] \\
&= \arg\min_\theta \int q(\mathbf{w}|\theta) \log \frac{q(\mathbf{w}|\theta)}{P(\mathbf{w})P(\mathcal{D}|\mathbf{w})} \mathrm{d}\mathbf{w} \\
&= \arg\min_\theta \underbrace{\boxed{\mathrm{KL}\left[q(\mathbf{w}|\theta) \;||\; P(\mathbf{w})\right]}}_{\text{prior-dependent complexity term}} - \underbrace{\boxed{\mathbb{E}_{q(\mathbf{w}|\theta)}\left[\log P(\mathcal{D}|\mathbf{w})\right]}}_{\text{Data-dependent Likelihood term}}
\end{aligned}
$$

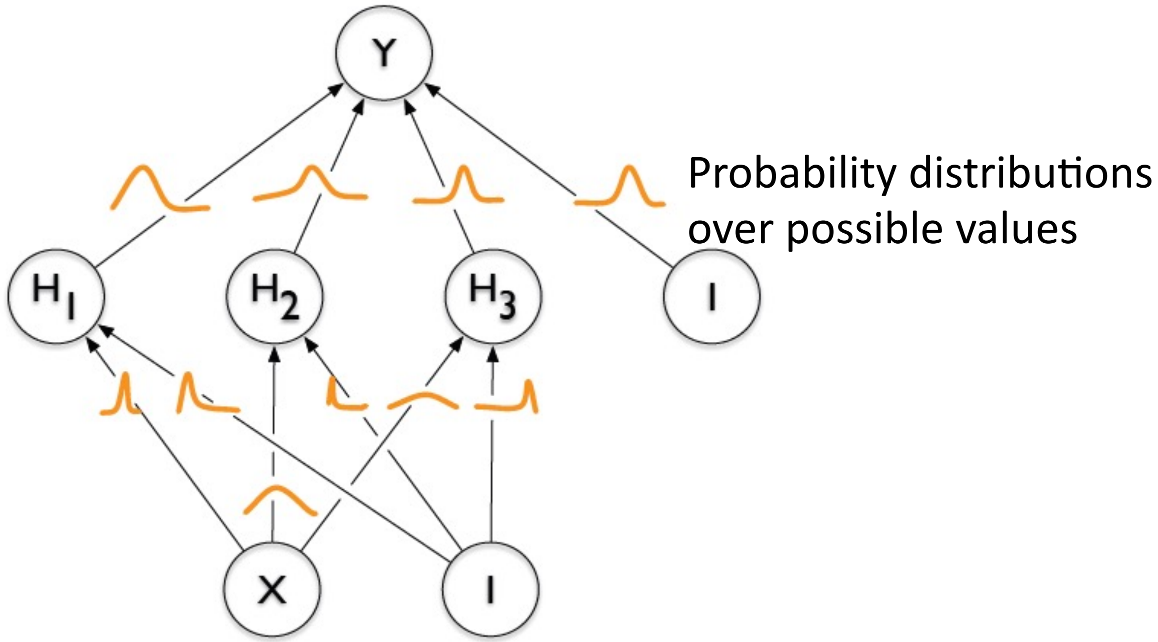
DG with Small Data

1️⃣ *Make the backbone more adaptive to small data scenarios*

Deterministic Neural Networks (e.g., CNNs, FCNs)



Bayesian Neural Networks

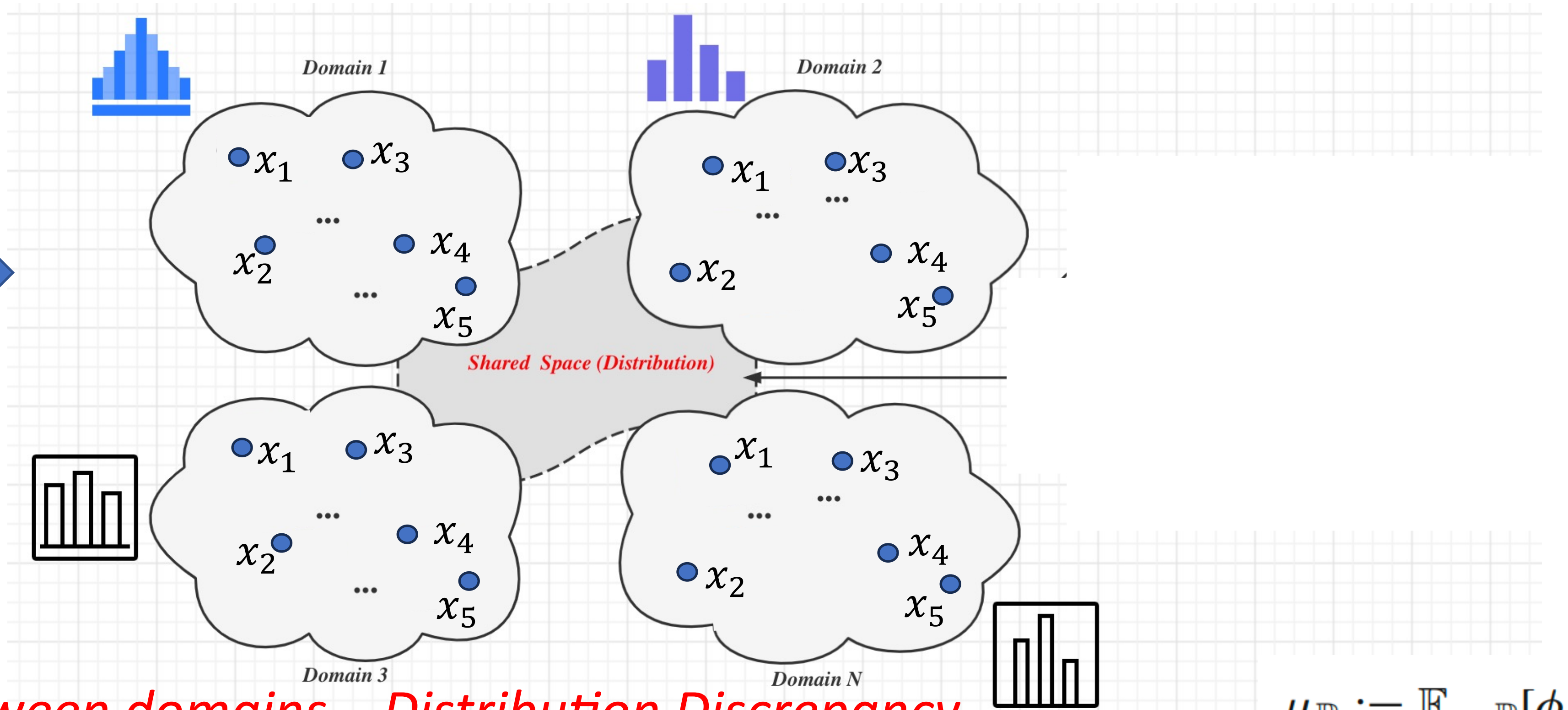


1. Richer representations and predictions from cheap model averaging.
2. Less overfitting risk in the context of small data scenarios

DG with Small Data

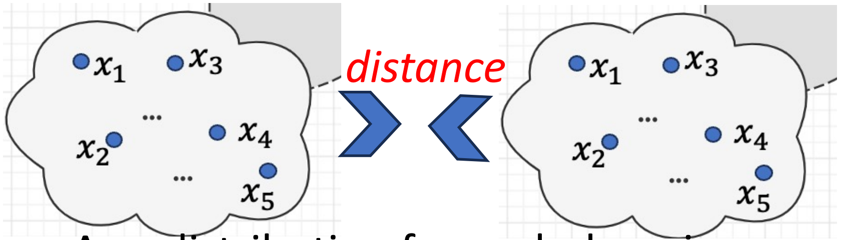
1️⃣ *Make the backbone more friendly to small data scenarios*

Resnet 18/50



Reduce the distance between domains – Distribution Discrepancy

$$\mu_{\mathbb{P}} := \mathbb{E}_{\mathbf{z}\sim\mathbb{P}}[\phi(\mathbf{z})]$$



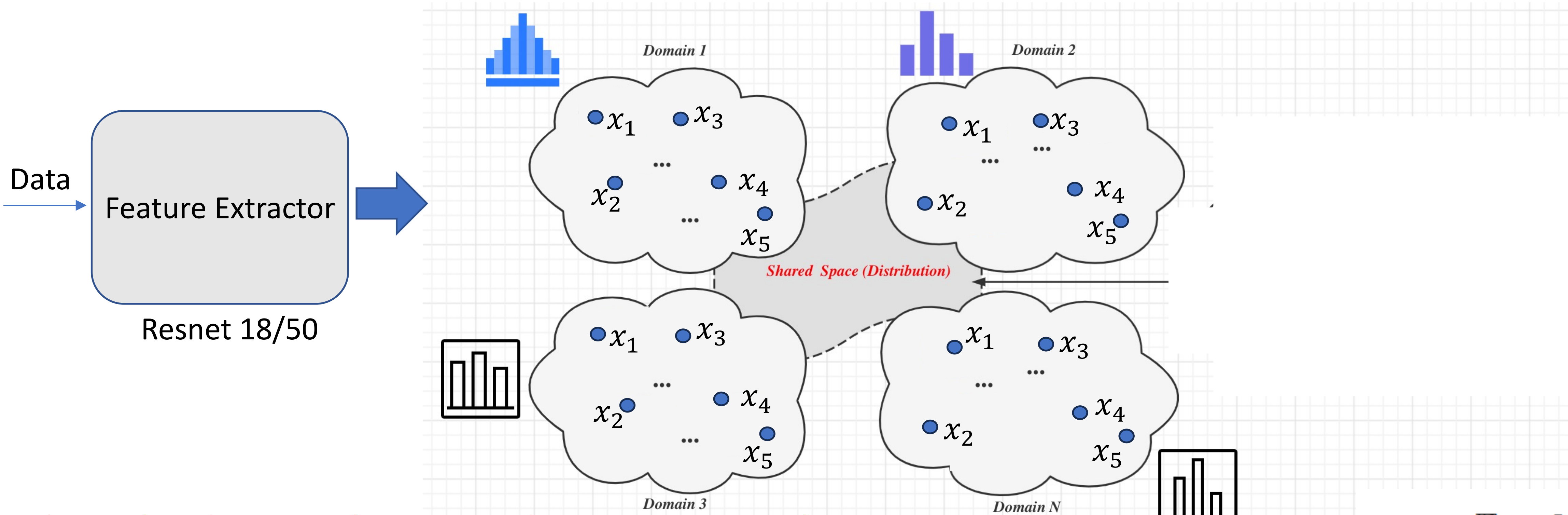
As a distribution for each domain

$$\operatorname{MMD}\left(\mathbb{P}_l, \mathbb{P}_t\right)^2 = \| \frac{1}{n_l} \sum_{i=1}^{n_l} \phi\left(\mathbf{z}_{l_i}\right) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi\left(\mathbf{z}_{t_j}\right) \|_{\mathcal{H}}^2$$

The probability measure can be mapped into a reproducing kernel Hilbert space (RKHS) as an element

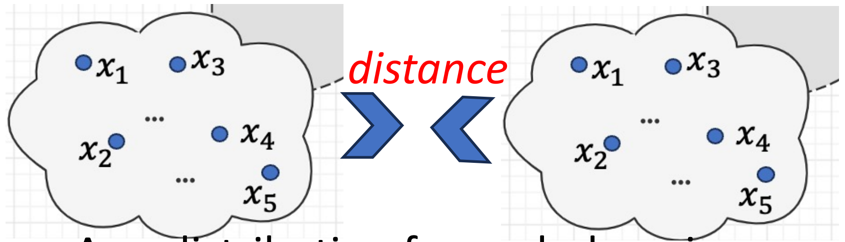
DG with Small Data

1️⃣ *Make the backbone more friendly to small data scenarios*



Reduce the distance between domains – Distribution Discrepancy

$$\mu_{\mathbb{P}} := \mathbb{E}_{\mathbf{z}\sim\mathbb{P}}[\phi(\mathbf{z})]$$



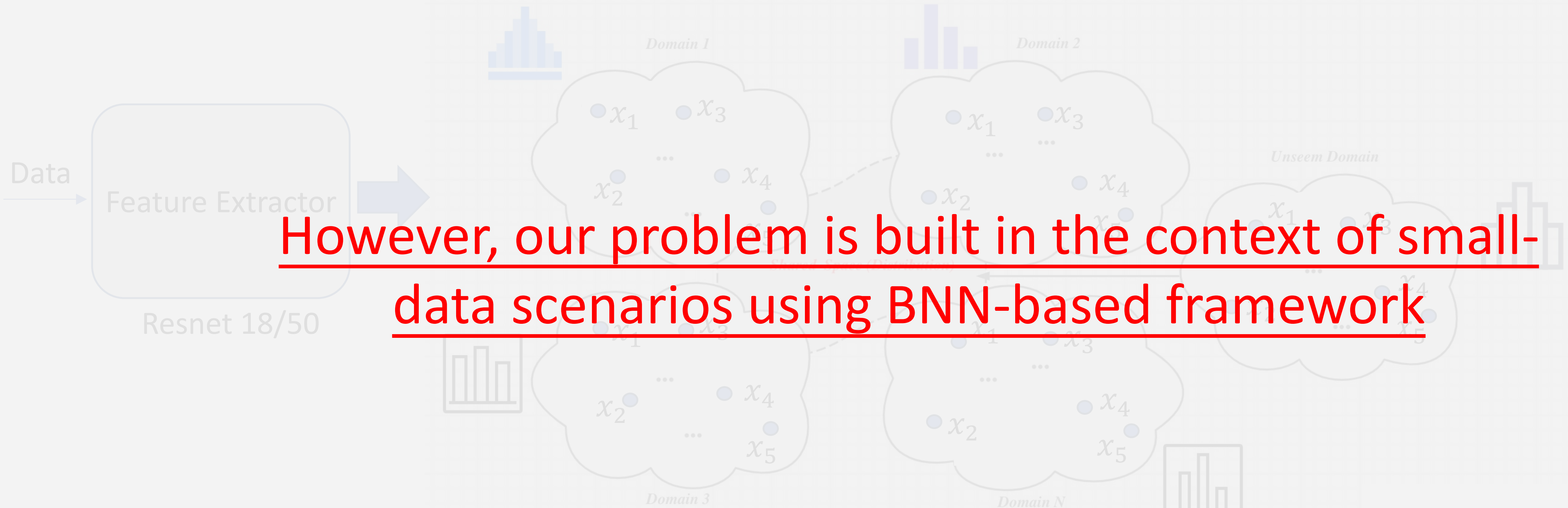
As a distribution for each domain

$$\mathrm{MMD}\left(\mathbb{P}_l, \mathbb{P}_t\right)^2 = \left\| \frac{1}{n_l} \sum_{i=1}^{n_l} \phi\left(\mathbf{z}_{l_i}\right) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi\left(\mathbf{z}_{t_j}\right) \right\|_{\mathcal{H}}^2$$

The probability measure can be mapped into a reproducing kernel Hilbert space (RKHS) as an element

DG with Small Data

1 *Make the backbone more friendly to small data scenarios*



However, our problem is built in the context of small-data scenarios using BNN-based framework

Reduce the distance between domains – Distribution Discrepancy

$$\mu_{\mathbb{P}} := \mathbb{E}_{\mathbf{z}\sim\mathbb{P}}[\phi(\mathbf{z})]$$

As a distribution for each domain

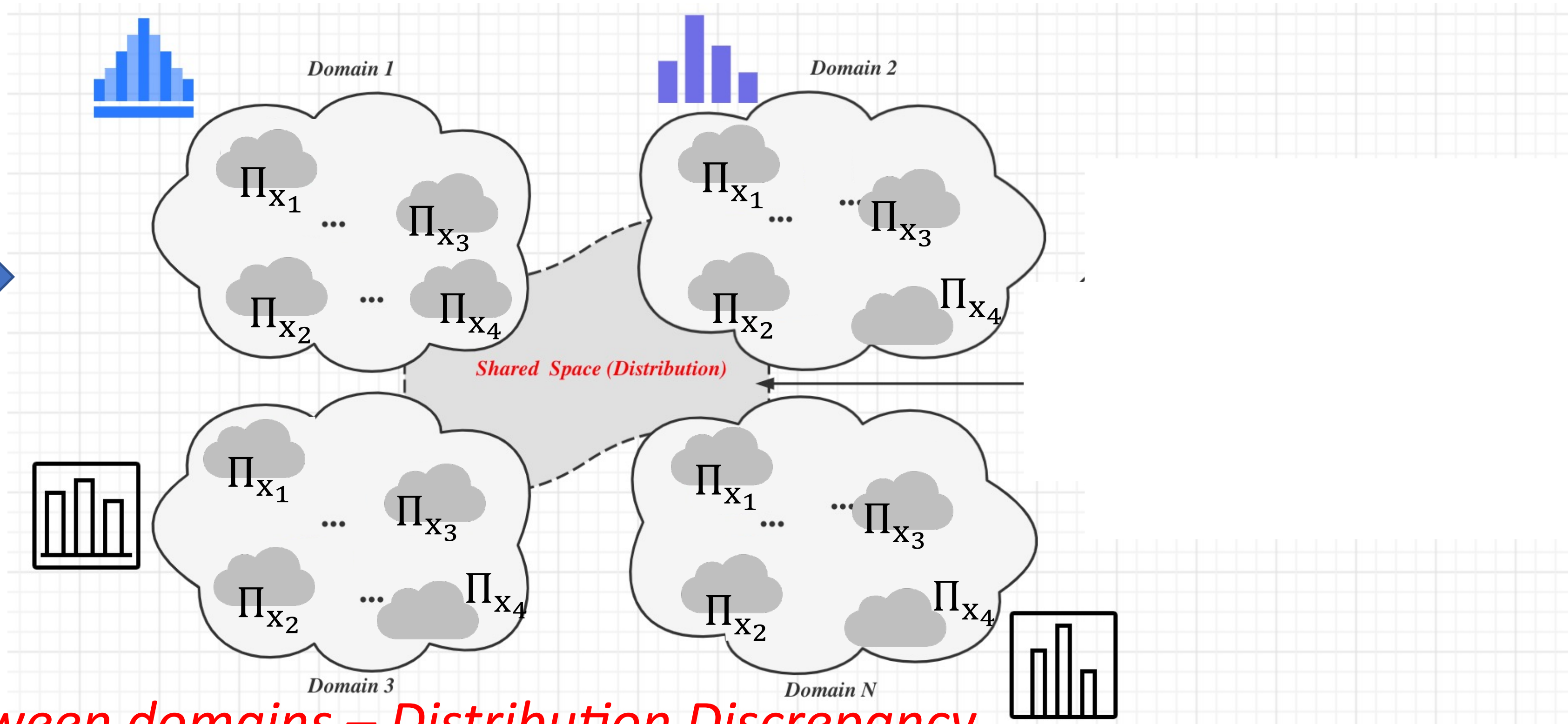
$$\mathrm{MMD}\left(\mathbb{P}_l, \mathbb{P}_t\right)^2 = \| \frac{1}{n_l} \sum_{i=1}^{n_l} \phi\left(\mathbf{z}_{l_i}\right) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi\left(\mathbf{z}_{t_j}\right) \|_{\mathcal{H}}^2$$

The probability measure can be mapped into a reproducing kernel Hilbert space (RKHS) as an element

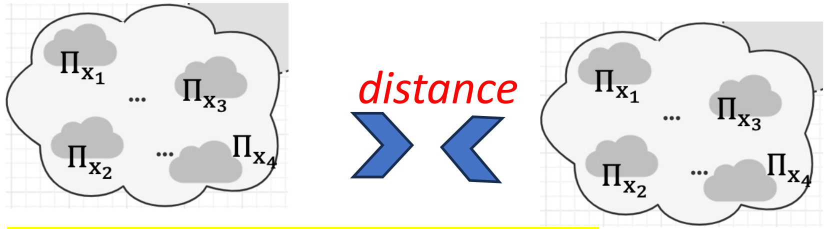
DG with Small Data

1️⃣ *Make the backbone more friendly to small data scenarios*

Data → BNN-based Feature Extractor



Reduce the distance between domains – Distribution Discrepancy



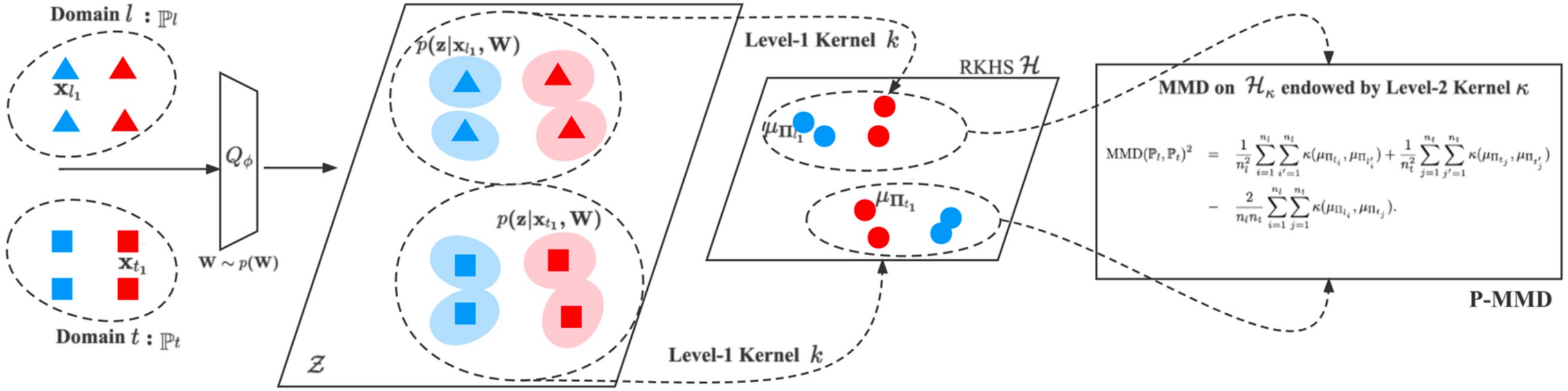
As a distribution over distributions for each domain

$$\mathbb{P}_l = \{\Pi_{l_1}, \ldots, \Pi_{l_{n_l}}\}$$

No previous works explore such distribution distance

DG with Small Data

2 *Probabilistic MMD (P-MMD) for the distribution over distributions*



Introduce a level-1 kernel κ and a level-2 kernel K

$$K(\Pi_{l_i}, \Pi_{t_j}) = \kappa(\mu_{\Pi_{l_i}}, \mu_{\Pi_{t_j}}) = \langle \psi(\mu_{\Pi_{l_i}}), \psi(\mu_{\Pi_{t_j}}) \rangle_{\mathcal{H}_\kappa}$$

Direct extension

$$\mathrm{MMD}(\mathbb{P}_l, \mathbb{P}_t)^2 = \| \frac{1}{n_l}\sum_{i=1}^{n_l}\phi(\mathbf{z}_{l_i}) - \frac{1}{n_t}\sum_{j=1}^{n_t}\phi(\mathbf{z}_{t_j}) \|_{\mathcal{H}}^2$$

$$\text{P-MMD}(\mathbb{P}_l, \mathbb{P}_t)^2 = \| \frac{1}{n_l}\sum_{i=1}^{n_l}\psi(\mu_{\Pi_{l_i}}) - \frac{1}{n_t}\sum_{j=1}^{n_t}\psi(\mu_{\Pi_{t_j}}) \|_{\mathcal{H}_\kappa}^2$$

$$= \frac{1}{n_l^2}\sum_{i=1}^{n_l}\sum_{i'=1}^{n_l} K(\Pi_{l_i}, \Pi_{l'_i}) + \frac{1}{n_t^2}\sum_{j=1}^{n_t}\sum_{j'=1}^{n_t} \boxed{K(\Pi_{t_j}, \Pi_{t'_j})}$$

$$- \frac{2}{n_l n_t}\sum_{i=1}^{n_l}\sum_{j=1}^{n_t} K(\Pi_{l_i}, \Pi_{t_j}).$$

$$K(\Pi_{l_i}, \Pi_{t_j}) = \kappa(\mu_{\Pi_{l_i}}, \mu_{\Pi_{t_j}}) = \exp(-\frac{\lambda}{2}\|\mu_{\Pi_{l_i}} - \mu_{\Pi_{t_j}}\|_{\mathcal{H}_\kappa}^2)$$

$$= \exp(-\frac{\lambda}{2}(\langle \mu_{\Pi_{l_i}}, \mu_{\Pi_{l_i}} \rangle_{\mathcal{H}_\kappa}) - 2\langle \mu_{\Pi_{l_i}}, \mu_{\Pi_{t_j}} \rangle_{\mathcal{H}_\kappa}$$

$$+ \langle \mu_{\Pi_{t_j}}, \mu_{\Pi_{t_j}} \rangle_{\mathcal{H}_\kappa}))$$

$$= \exp(-\frac{\lambda}{2}(\frac{1}{m_l^2}\sum_{i=1}^{m_l}\sum_{i'=1}^{m_l} k(\mathbf{z}_{l_i}, \mathbf{z}_{l'_i})$$

$$- \frac{2}{m_l m_t}\sum_{i=1}^{m_l}\sum_{j=1}^{m_t} k(\mathbf{z}_{l_i}, \mathbf{z}_{t_j})) + \frac{1}{m_t^2}\sum_{j=1}^{m_t}\sum_{j'=1}^{m_t} k(\mathbf{z}_{t_j}, \mathbf{z}_{t'_j}),$$

Unbiased estimation

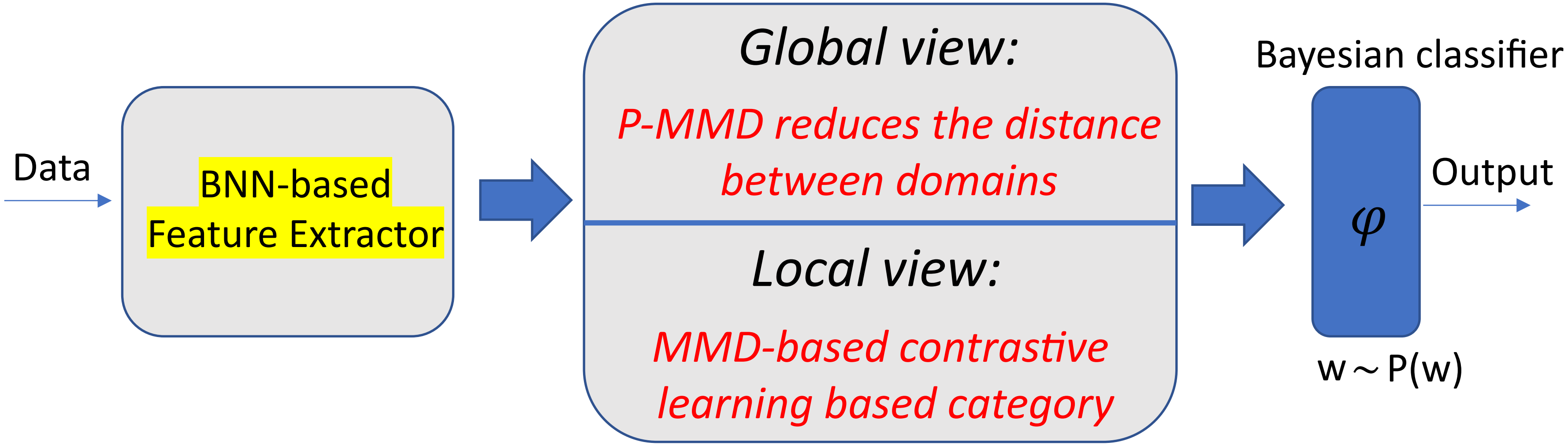
Reduce computation complexity from O(n2) => O(n)

$$\text{P-MMD}(\mathbb{P}_l, \mathbb{P}_t)^2 \approx \frac{2}{n_l}\sum_{i=1}^{\frac{2}{n_l}}[K(\Pi_{l_{2i}}, \Pi_{l'_{2i+1}}) + K(\Pi_{t_{2i}}, \Pi_{t'_{2i+1}}) - K(\Pi_{l_{2i}}, \Pi_{t_{2i+1}}) - K(\Pi_{l_{2i+1}}, \Pi_{t_{2i}})$$

Hint: $\mu_{\mathbb{P}} := \mathbb{E}_{\mathbf{z}\sim\mathbb{P}}[\phi(\mathbf{z})]$ An element on RKHS

Unbiased estimation: Drawing pairs of data from two domains with replacement

DG with Small Data

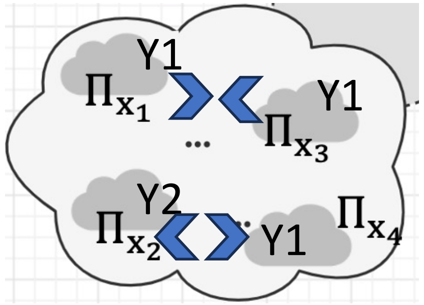


Learn shared feature space

Local view:

$$\mathcal{L}_{local}^{pos} = \frac{1}{2}\| \frac{1}{T}\sum_{i=1}^{T}\phi\left(M_\Theta(\mathbf{z}_{n_i})\right) - \frac{1}{T}\sum_{j=1}^{T}\phi\left(M_\Theta(\mathbf{z}_{q_j})\right)\|_{\mathcal{H}}^2,$$

Same category



$$\mathcal{L}_{local}^{neg} = \frac{1}{2}\max[0, \xi - \mathrm{MMD}(\Pi_n, \Pi_q)^2] = \frac{1}{2}\max[0, \xi - \| \frac{1}{T}\sum_{i=1}^{T}\phi\left(M_\Theta(\mathbf{z}_{n_i})\right) - \frac{1}{T}\sum_{j=1}^{T}\phi\left(M_\Theta(\mathbf{z}_{q_j})\right)\|_{\mathcal{H}}^2],$$

Different category

DG with Small Data

Performance on skin lesion classification task

Table 2 Domain generalization results on skin lesion classification

Method	DMF	D7P	MSK	PH2	SON	UDA	Average
DeepAll	0.2492 ±0.0127	0.5680±0.0181	0.6674±0.0083	0.8000±0.0167	0.8613±0.0296	0.6264±0.0312	0.6287
MASF (Dou et al., 2019)	0.2692±0.0146	0.5678±0.0361	0.6815±0.0122	0.7833±0.0101	0.9204±0.0227	0.6538±0.0196	0.6460
LDDG (Li et al., 2020)	0.2793±0.0244	0.6007±0.0187	0.6967±0.0211	0.8167±0.0209	0.9272±0.0117	0.6978±0.0182	0.6697
KDDG (Wang et al., 2021)	0.3189±0.0256	0.5829±0.0212	0.7014±0.0178	0.9021±0.0314	0.9398±0.0213	0.6882±0.0139	0.6889
SWAD (Cha et al., 2021)	0.3582 ±0.0234	0.5491 ±0.0231	0.6842 ±0.0156	0.9167 ±0.0121	0.9824 ±0.0012	0.7240 ±0.0251	0.7024
BDIL (Xiao et al., 2021)	0.2985±0.0452	**0.6204**±0.0212	0.7059±0.0145	0.8967±0.0096	<u>0.9860</u>±0.0198	0.7219±0.0284	0.7049
DNA (Chu et al., 2022)	0.3532 ±0.0133	0.5581 ±0.0178	<u>0.7120</u> ±0.0194	<u>0.9333</u>±0.0045	0.9851 ±0.0032	0.7314 ±0.0141	0.7122
DSU (Li et al., 2022)	**0.3830** ±0.0267	0.5739 ±0.0147	0.6935 ±0.0165	0.8833 ±0.0231	0.9841 ±0.0098	0.7201 ±0.0121	0.7063
MIRO (Cha et al., 2022)	0.3432 ±0.0092	0.5863 ±0.0113	0.6919 ±0.0101	0.9300±0.0021	0.9659 ±0.0292	<u>0.7328</u> ±0.0233	0.7084
Ours (in this paper)	<u>0.3781</u>±0.0136	<u>0.6120</u>±0.0115	**0.7276** ±0.0201	**0.9416**±0.0103	**0.9889**±0.0041	**0.7486** ±0.0123	**0.7328**

Each column denotes a cross-domain task. For example, in the second column, we use DMF dataset as the target domain and the remaining datasets as the source domains. The best and second-best performance on each target domain are bolded and underlined, respectively. Note that all baseline methods adopt the SWAD method (Cha et al., 2021) for weight averaging. The baseline in the sixth row, namely SWAD, denotes the ERM training strategy with the SWAD method

DG with Small Data

Performance on gray matter segmentation task

Table 3 Domain generalization results on gray matter segmentation task. For the DSC, CC, TPR, and JI, the higher the better. For the ASD, the lower the better. Note that all baseline methods adopt the SWAD method (Cha et al., 2021) for weight averaging. The baseline, namely SWAD, denotes the ERM training strategy with the SWAD method.

(a) MASF

source	target	DSC	CC	JI	TPR	ASD
2,3,4	1	0.8502	64.22	0.7415	0.8903	0.2274
1,3,4	2	0.8115	53.04	0.6844	0.8161	0.0826
1,2,4	3	0.5285	-99.3	0.3665	0.5155	1.8554
1,2,3	4	**0.8938**	**76.14**	**0.8083**	<u>0.8991</u>	0.0366
Average		0.7710	23.52	0.6502	0.7803	0.5505

(b) KDDG

source	target	DSC	CC	JI	TPR	ASD
2,3,4	1	<u>0.8745</u>	<u>70.75</u>	<u>0.7795</u>	0.8949	0.0539
1,3,4	2	0.8229	56.71	0.6997	0.8226	0.0490
1,2,4	3	**0.5676**	**-63.1**	0.3866	0.5904	<u>1.2805</u>
1,2,3	4	0.8894	75.06	0.8011	0.9222	0.0377
Average		0.7886	34.86	0.6667	0.8075	0.3553

(c) LDDG

source	target	DSC	CC	JI	TPR	ASD
2,3,4	1	0.8708	69.29	0.7753	0.8978	**0.0411**
1,3,4	2	0.8364	60.58	0.7199	<u>0.8485</u>	0.0416
1,2,4	3	0.5543	-71.6	<u>0.3889</u>	<u>0.5923</u>	1.5187
1,2,3	4	0.8910	75.46	0.8039	0.8844	**0.0289**
Average		0.7881	33.43	0.6720	0.8058	0.4076

(d) SWAD

source	target	DSC	CC	JI	TPR	ASD
2,3,4	1	0.8726	70.23	0.7702	0.8995	0.0502
1,3,4	2	0.8378	60.71	0.7230	0.8176	0.0424
1,2,4	3	0.5388	-99.0	0.3789	0.5083	1.4789
1,2,3	4	0.8903	<u>75.89</u>	0.8026	0.8859	<u>0.0302</u>
Average		0.7849	26.96	0.6687	0.7778	0.4002

(e) DSU

source	target	DSC	CC	JI	TPR	ASD
2,3,4	1	0.8739	70.32	0.7794	<u>0.9210</u>	0.0793
1,3,4	2	0.8474	63.58	0.7367	**0.8502**	0.0494
1,2,4	3	0.5574	-70.4	0.3923	0.6097	1.5049
1,2,3	4	0.8897	75.10	0.8018	0.9225	0.0415
Average		0.7921	34.65	0.6775	0.8225	0.4362

(f) Ours

source	target	DSC	CC	JI	TPR	ASD
2,3,4	1	**0.8786**	**71.57**	**0.7873**	**0.9293**	<u>0.0422</u>
1,3,4	2	**0.8485**	**63.78**	**0.7389**	0.8401	**0.0401**
1,2,4	3	<u>0.5634</u>	<u>-68.0</u>	**0.3992**	**0.6103**	**1.2239**
1,2,3	4	<u>0.8921</u>	75.69	<u>0.8058</u>	**0.9245**	0.0362
Average		**0.7957**	**35.76**	**0.6828**	**0.8260**	**0.3356**

DG with Small Data

Performance on extremely small-data scenarios

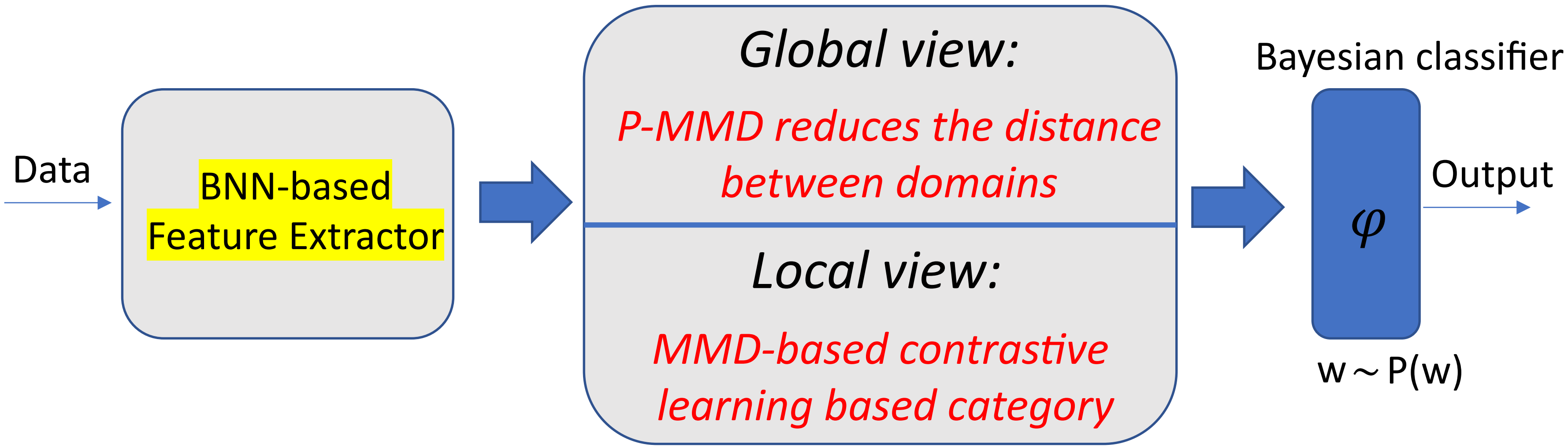
Table 5 Domain generalization results on MSK dataset by randomly picking same proportion of samples from each source domain

Proportion (%)	BDIL	DNA	Ours
100	0.7059± 0.0284	0.7121± 0.0141	**0.7276**±0.0123
80	0.6625±0.0920	0.6591±0.0022	**0.6975**±0.0036
60	0.6468±0.0106	0.6149±0.0112	**0.6641**±0.0114
40	0.6491±0.0171	0.6065±0.0111	**0.6579**±0.0057
Average (80,60,40) ↑	0.6528	0.6268	**0.6732**
Average Attenuation Rate ↓	7.67%	11.98%	**7.37%**

A smaller proportion ($< 40\%$) is unavailable because equal batch sizes cannot be maintained in PH2 dataset

Table 6 Domain generalization results on MSK dataset by randomly picking same number of samples from each class in each domain

Number of sample	BDIL	DNA	Ours
40	0.5897 ± 0.0029	0.5412 ± 0.0143	**0.6368** ± 0.0074
30	0.5762 ± 0.0101	0.5132 ± 0.0229	**0.6138** ± 0.0291
20	0.5573 ± 0.0011	0.5048 ± 0.0087	**0.6037** ± 0.0121
Average (40,30,20) ↑	0.5744	0.5196	**0.6183**
Average attenuation rate ↓	5.49%	6.72%	**5.19%**



SDDG:

1. BNN is more adaptive to small-data scenarios.

2. A new extension of MMD is proposed to compute the distribution distance between distributions over distributions.

3. A more generalized model can be learned by DG in the context of small data

Thanks !