Continuous Domain Generalization

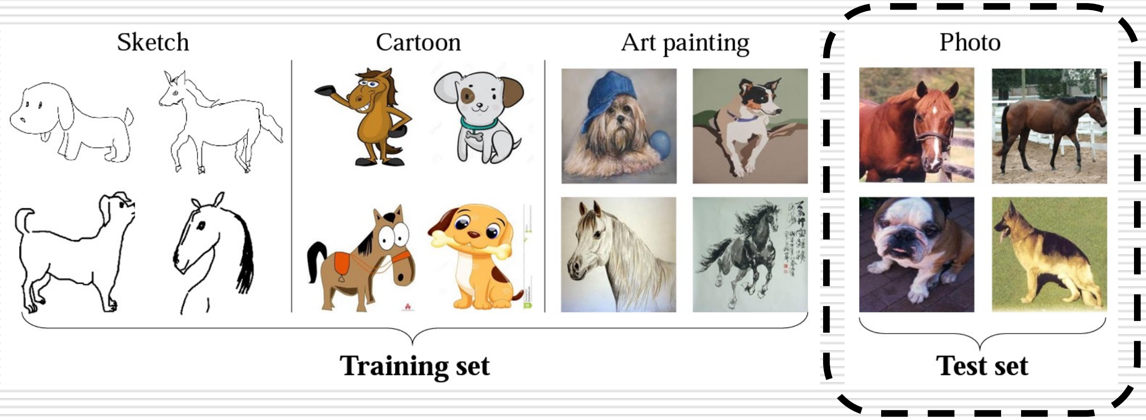
Zekun Cai[1, 4], Yiheng Yao[2], Guangji Bai[3], Renhe Jiang[2], Xuan Song[1], Ryosuke Shibasaki[4], Liang Zhao[3]

1 Jilin University, Changchun, China
2 The University of Tokyo, Tokyo, Japan
3 Emory University, Atlanta, GA, USA
4 LocationMind, Tokyo, Japan

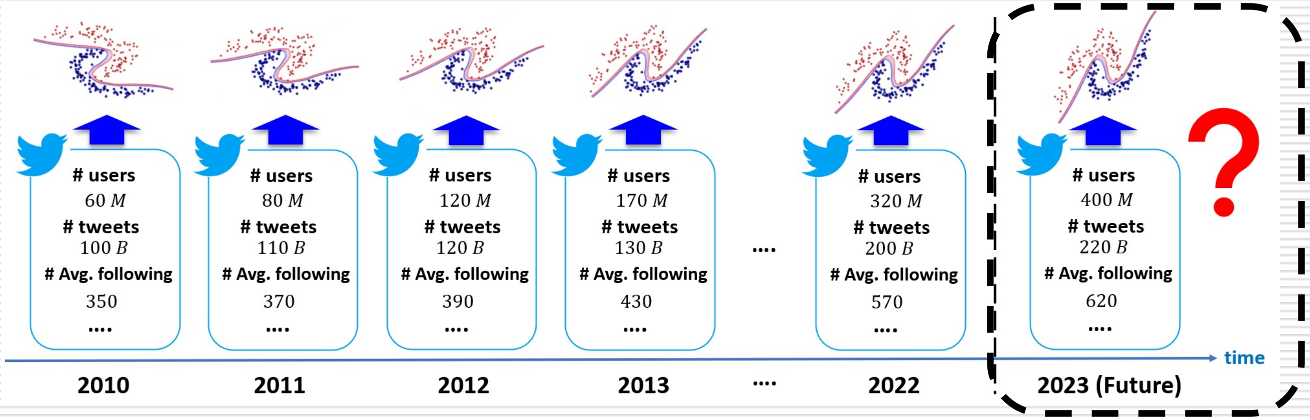
Introduction

- Domain Generalization (DG) and Temporal Domain Generalization (TDG)
  - In practice, the distribution of training data differs from that of test data.

DG Out-of-distribution (OOD)



TDG Out-of-distribution (OOD)



Categorical Entities with No Time Dependency

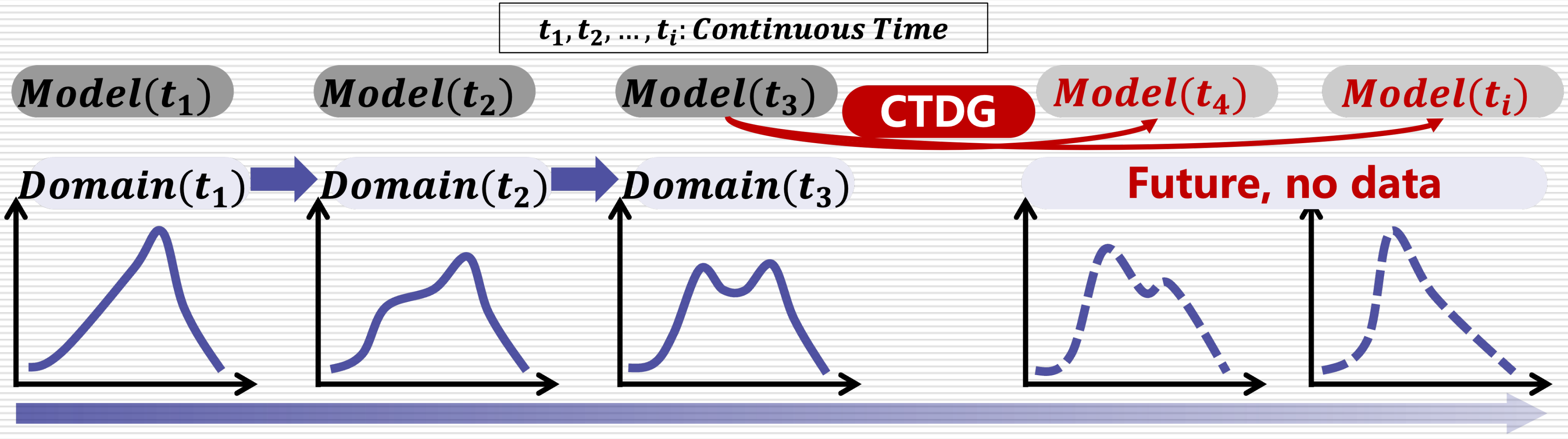
DG: To learn a generalized model that performs well on the unseen target domain.

Temporal Sequence but the domain is always regularly distributed

TDG: Domain shifts are temporally correlated. It extends DG approaches by modeling domains as a sequence.

Wang, Jindong, et al. "Generalizing to unseen domains: A survey on domain generalization." IEEE TKDE (2022).
Bai, Guangji, et al. "Temporal Domain Generalization with Drift-Aware Dynamic Neural Networks." ICLR (2023).

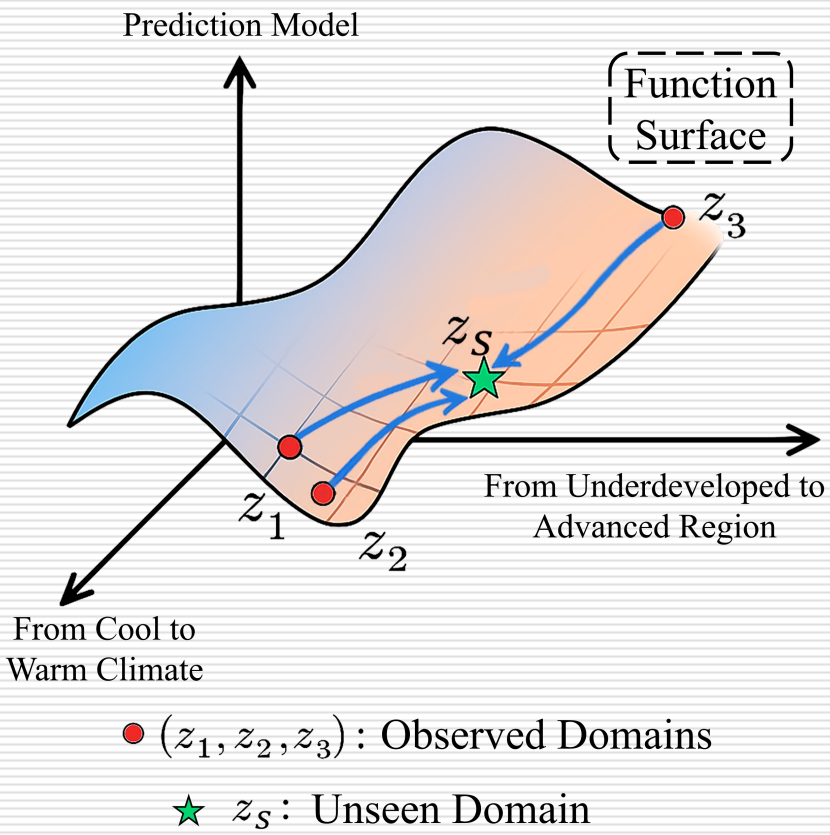
Extending One-dimensional Temporal Generalization to Generalization in Arbitrary-dimensional Spaces



CTDG: Evolution in a single dimension

Cai Zekun, et al. "Continuous Temporal Domain Generalization." NeurIPS (2024).

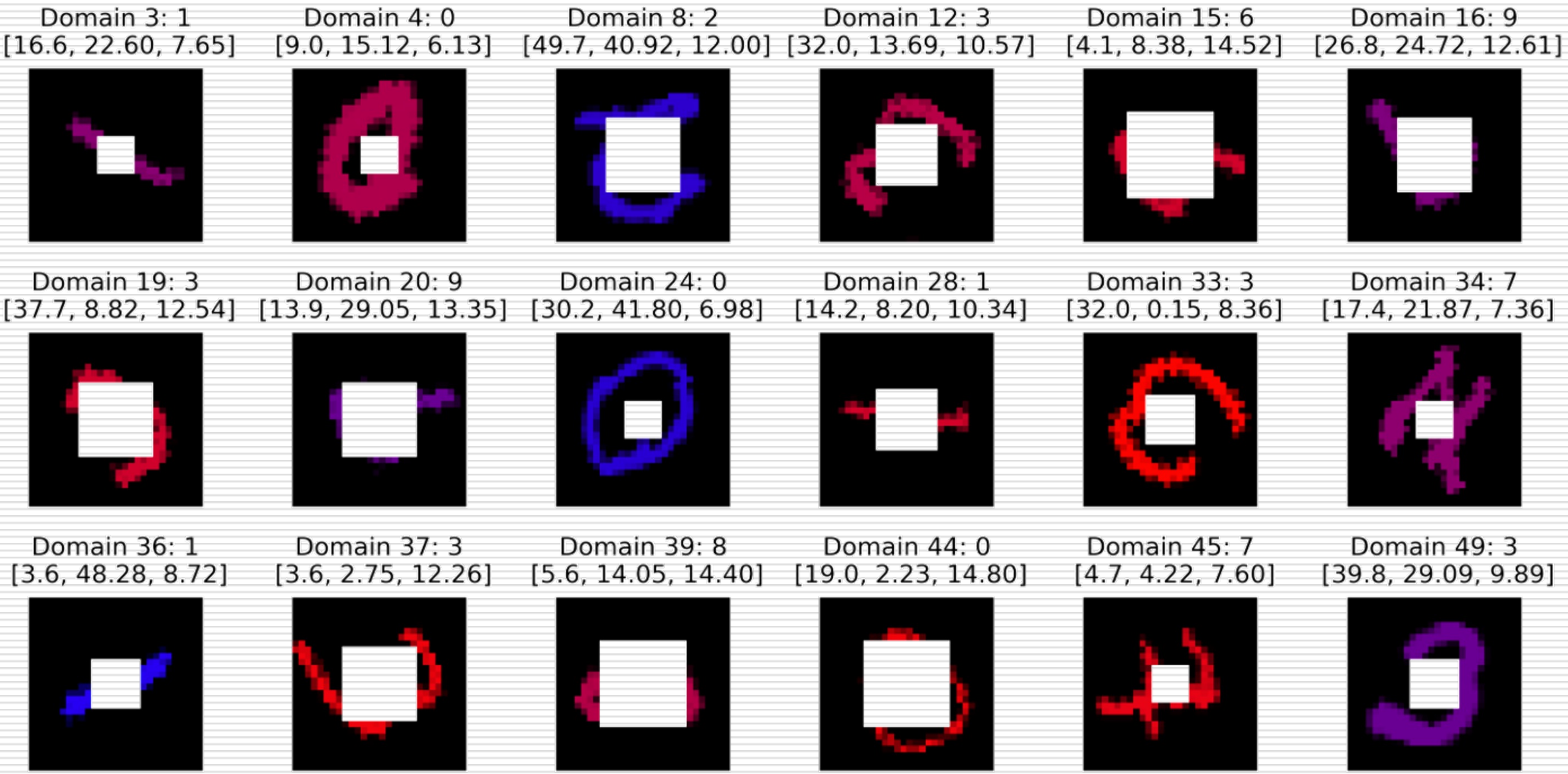
Extending One-dimensional Temporal Generalization to Generalization in Arbitrary-dimensional Spaces



- Data distribution is influenced by multiple continuous factors (e.g., climate, economic level, geographical location, urban structure).
- The relationships between domains are continuous, not discrete and abrupt.
- Therefore, a domain is a point in a continuous space.

CDG: Evolution in any dimension

Extending One-dimensional Temporal Generalization to Generalization in Arbitrary-dimensional Spaces



Each domain corresponds to a unique combination of three transformations: rotation, color shift, and occlusion.
One representative image is sampled from each domain for visualization.
Can you correctly identify the image?

Neural network error rate: 31% ， BAD!

Critical Hurdles

- **(Theoretical Part)** How to identify domain variation to model evolution in general spaces?
  - Solution: Proof **Parameter Submanifold Theorem**
- **(Structural Part)** How to model without chronological structure domains?
  - Solution: Proposed to integrate **Lie Group theory** with a neural transport operator
- **(Practical Part)** How to learn under imperfect representations of domain variation?
  - Solution: Proposed gating mechanism and a local chart-based strategy

Solutions 1/3: Identifying the Parameter Manifold of Continuous Domains

- **Key Assumption: Distribution Continuity**
- We assume that data distributions evolve continuously over time:

- **Key Theorem: Model Continuity**
- It can be proved that the model parameters are also continuously evolving:

Advantages

- ***Unified Representation***
- ***Geometric Regularization***
- ***Analytical Operability***

- **There is a theoretical space that provides a complete and non-degenerate representation of all latent factors governing domain variation**
- If the dimension of such space is d
- The model parameter dimension is D
- The collection of domain-wise parameters forms a d -dimensional **embedded submanifold** of $\mathbb{R}^D$.

Solutions 2/3: Neural Domain Transport Operator under Structural Constraints

Can we replace ODE used in TDG, with PDE? It is really challenge!

We do not have an analytical formula of PDE for the distribution variation.

Definition 1 (Geometric Structure). *The neural transport operator $\mathcal{T}$ is said to satisfy geometric structure if it is continuous in all of its inputs.*

Definition 2 (Algebraic Structure). *The neural transport operator $\mathcal{T}$ is said to satisfy algebraic structure if the following properties hold:*

- ***Closure:*** $\mathcal{T}(\theta(z_i), z_i, z_j) \in \Theta$, *ensuring transported parameters remain within the valid space.*
- ***Identity:*** $\mathcal{T}(\theta(z_i), z_i, z_i) = \theta(z_i)$, *ensuring that self-transport leaves parameters unchanged.*
- ***Associativity:*** $\mathcal{T}(\mathcal{T}(\theta(z_i), z_i, z_j), z_j, z_k) = \mathcal{T}(\theta(z_i), z_i, z_k)$, *ensuring that sequential transports are equivalent to direct ones.*
- ***Invertibility:*** $\mathcal{T}^{-1}\mathcal{T}(\theta(z_i), z_i, z_j) = \theta(z_i)$, *ensuring that each transport can be exactly reversed.*

- **Geometric Structure -> Manifold**
- **Algebraic Structure -> Group**
- **Manifold + Group -> Lie Group**

In mathematics, a Lie group is a group that is also a differentiable manifold, such that group multiplication and taking inverses are both differentiable.

Solutions 3/3: Handling Imperfect Descriptors Space

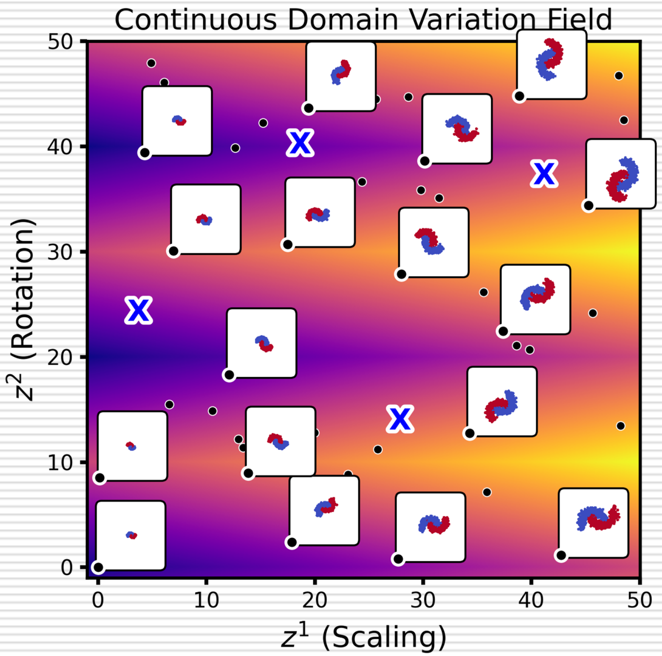
- **The Manifold Theorem discusses a theoretically grounded descriptor space.**
- **In the real world, however, the descriptors accessible for model training are observable projections of this theoretical space**

- Degeneracy
  - Some directions in the space fail to affect the distribution.
- Incompleteness
  - The space partially capture the degrees of freedom governing domain shifts

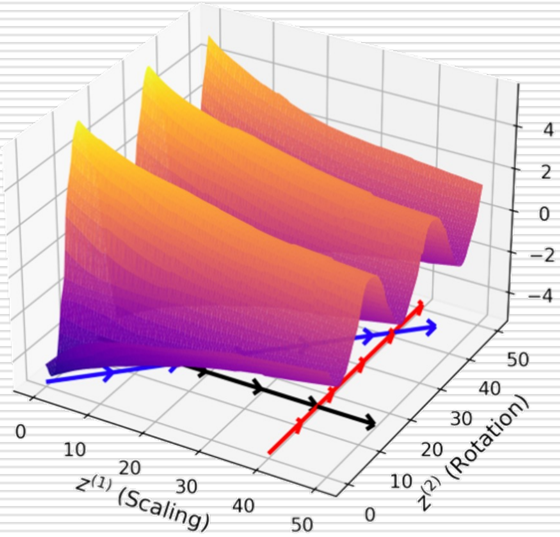
- Suppressing Degeneracy via Descriptor Gating
  - We apply a dimension-wise gate to modulate the influence of each dimension.
- Mitigating Incompleteness via Local Chart
  - The parameter manifold is represented as an atlas, i.e., a collection of overlapping local charts. Transfer only be conducted within each chart.

Ideas from differential geometry.

Synthetic Dataset Results: Generalize Model to Any Space



All training domains (black dots), with several illustrated in small inset panels.



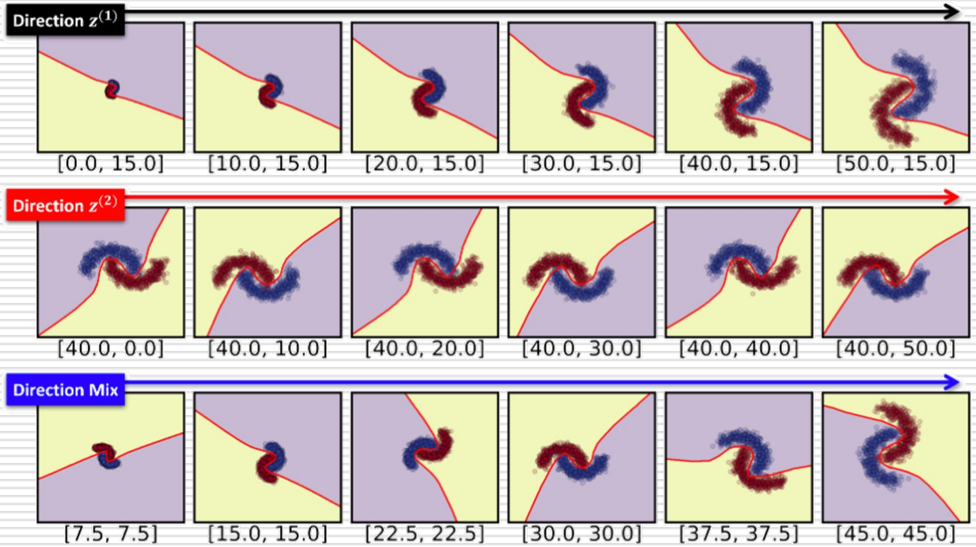


Figure 3: Visualization of the learned parameter manifold and the corresponding generalization behavior. **Left:** PCA projection of predicted parameters $\theta(z)$ over the entire descriptor space. **Right:** Visualization of decision boundaries and data samples along selected direction.

Quantitative Results Analysis

Table 1: Performance comparison on continuous domain datasets. Classification tasks report error rates (%) and regression tasks report MAE. 'N/A' implies that the method does not support the task.

Model	Classification					Regression
	2-Moons	MNIST	fMoW	ArXiv	Yearbook	Traffic
Descriptor-Agnostic						
ERM	34.7 ± 0.2	31.8 ± 0.9	27.7 ± 1.6	35.6 ± 0.1	8.6 ± 1.0	16.4 ± 0.1
IRM [1]	34.4 ± 0.2	33.0 ± 0.8	41.5 ± 2.8	37.4 ± 1.0	8.3 ± 0.5	16.6 ± 0.1
V-REx [26]	34.9 ± 0.1	32.2 ± 1.4	32.1 ± 3.6	37.3 ± 0.7	8.9 ± 0.5	20.9 ± 0.6
GroupDRO [46]	34.5 ± 0.1	37.6 ± 1.0	28.6 ± 1.9	35.6 ± 0.1	8.0 ± 0.4	16.2 ± 0.1
Mixup [54]	34.9 ± 0.1	34.0 ± 0.9	27.1 ± 1.5	35.5 ± 0.2	7.5 ± 0.5	16.1 ± 0.1
DANN [17]	35.1 ± 0.4	34.7 ± 0.6	26.0 ± 0.7	36.5 ± 0.2	8.9 ± 1.4	18.1 ± 0.2
MLDG [27]	34.6 ± 0.2	85.1 ± 2.5	29.2 ± 1.0	35.8 ± 0.2	7.7 ± 0.5	16.9 ± 0.1
CDANN [32]	35.0 ± 0.2	36.4 ± 0.8	27.6 ± 0.9	36.2 ± 0.2	8.7 ± 0.4	17.3 ± 0.2
URM [25]	34.7 ± 0.1	31.8 ± 1.3	26.9 ± 1.0	35.5 ± 0.4	8.0 ± 0.3	16.2 ± 0.2
Descriptor-Aware						
ERM-D	13.1 ± 1.5	31.7 ± 0.5	28.9 ± 1.8	38.1 ± 0.6	7.4 ± 0.5	15.9 ± 0.1
NDA	25.4 ± 0.3	26.3 ± 0.7	31.2 ± 1.4	35.6 ± 0.6	11.0 ± 0.8	17.2 ± 0.2
CIDA [51]	14.2 ± 1.1	27.4 ± 0.5	27.1 ± 0.9	35.3 ± 0.4	8.4 ± 0.8	16.6 ± 0.1
TKNets [59]	N/A	N/A	N/A	N/A	8.4 ± 0.3	N/A
DRAIN [2]	N/A	N/A	N/A	N/A	10.5 ± 1.0	N/A
Koodos [6]	N/A	N/A	N/A	N/A	6.6 ± 1.3	N/A
NeuralLio (Ours)	**3.2 ± 1.2**	**9.5 ± 1.1**	**24.5 ± 0.5**	**34.7 ± 0.4**	**4.8 ± 0.3**	**15.1 ± 0.1**

Thank you for your attention.

- Paper: https://arxiv.org/abs/2505.13519
- Contact: caizekun@csis.u-tokyo.ac.jp