The 38th Annual AAAI Conference on Artificial Intelligence
FEBRUARY 20–27, 2024 | VANCOUVER, CANADA
VANCOUVER CONVENTION CENTRE – WEST BUILDING

PDE+: Enhancing Generalization via PDE with Adaptive Distributional Diffusion

Yige Yuan[1,2], Bingbing Xu[1], Bo Lin[3], Liang Hou[1,2], Fei Sun[1], Huawei Shen[1,2], Xueqi Cheng[1,2]

[1]CAS Key Laboratory of AI Safety and Security, Institute of Computing Technology, Chinese Academy of Sciences

[2]University of Chinese Academy of Sciences

[3]Department of Mathematics, National University of Singapore

Paper

Code

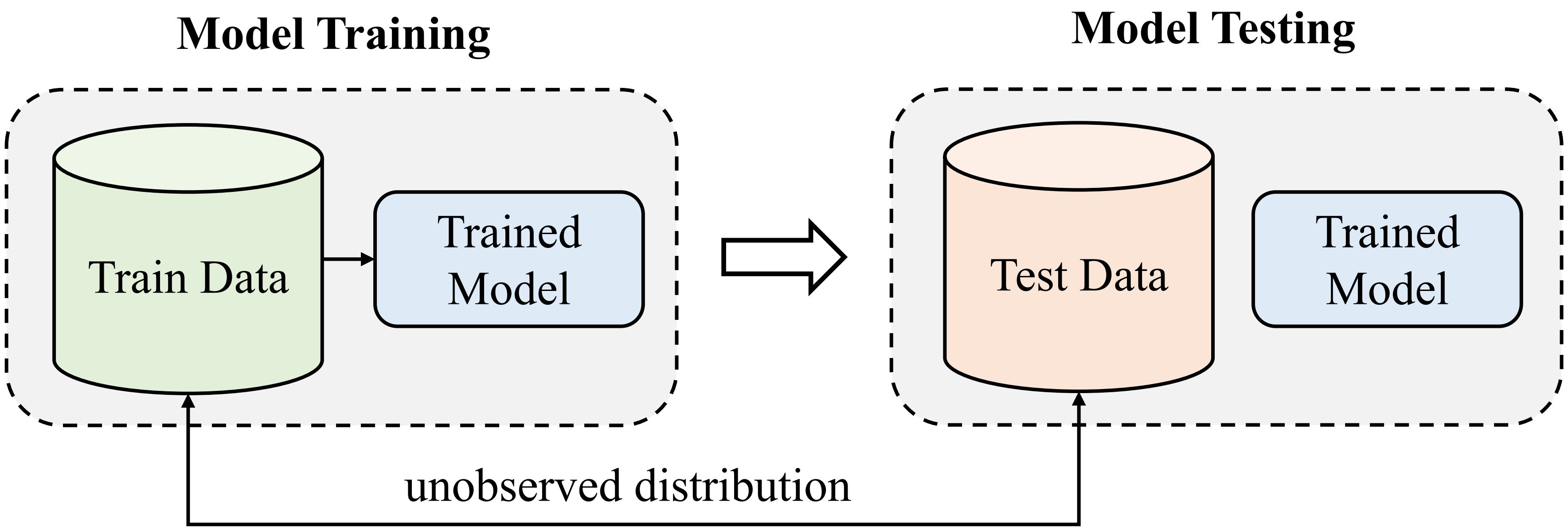
Presenter: Yige Yuan
Email: yuanyige20z@ict.ac.cn

- **Motivation**
- **Method**
- **Experiments**

MOTIVATION

Generalization

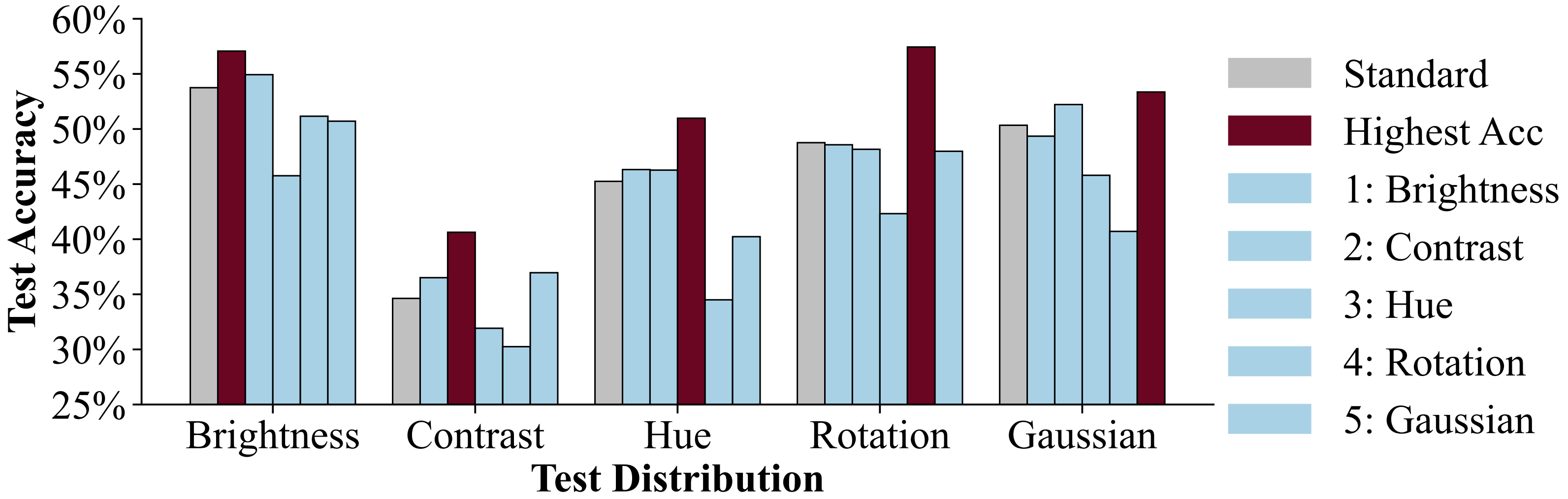
- Generalization is the ability of trained neural networks to perform effectively under **unobserved** distributions.



MOTIVATION

Existing Work

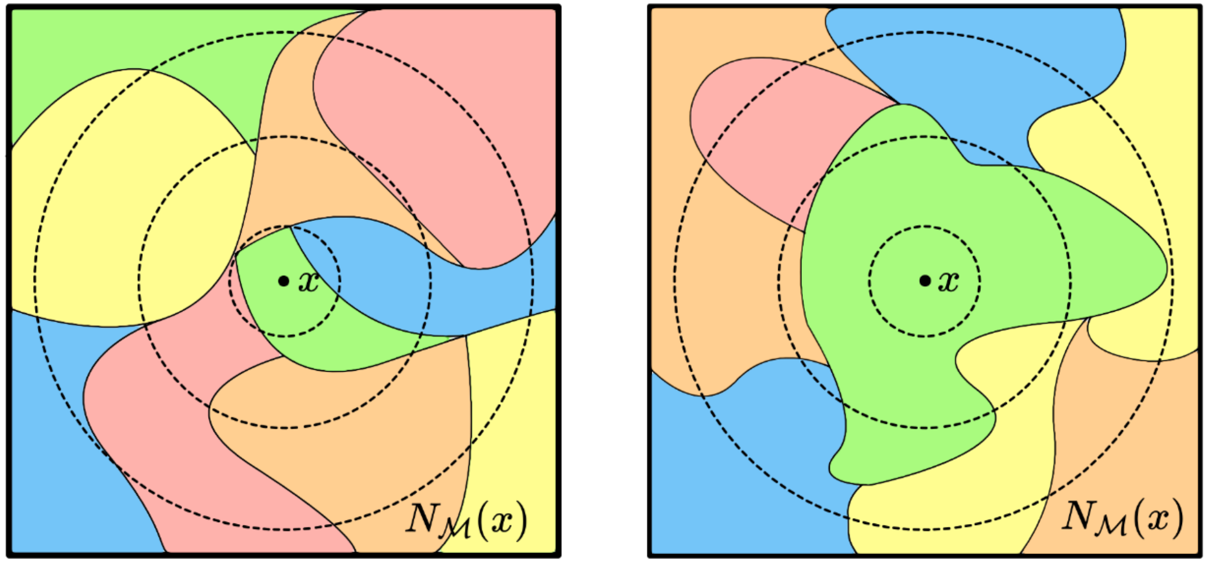
- Existing data-driven paradigm usually cannot guarantee reliable generalization capabilities on unobserved distributions.
  - **Data Augmentation**: Mixup, AutoAug, AugMix, etc.
  - **Adversarial Training**: PGD, RLAT, etc.
  - **Noise Injection**: RSE, NFM, EnResNet, etc.



MOTIVATION

Existing Work

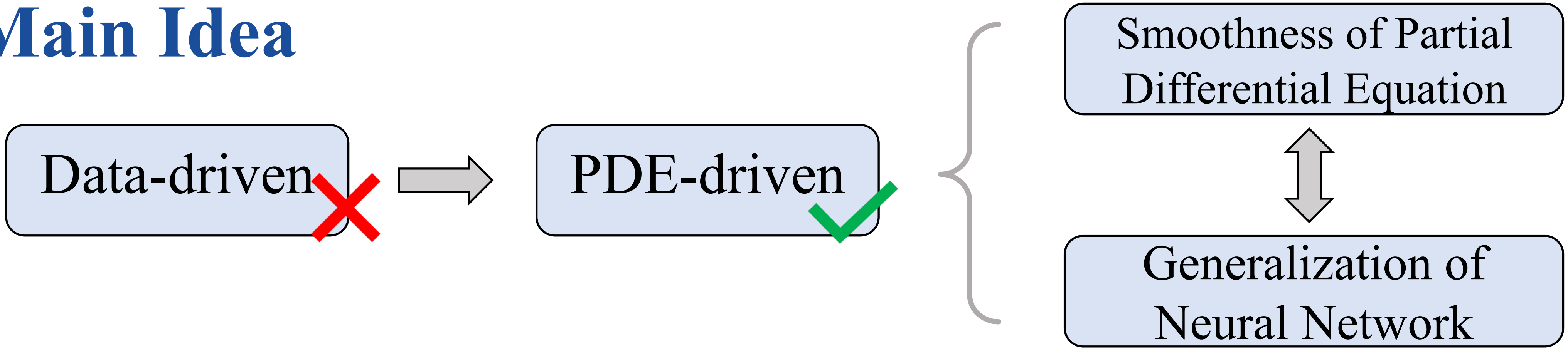
- The limited generalization capabilities of data-driven paradigm is due to model irregularity (non-smoothness).



Less Smooth Classifier   More Smooth Classifier

MOTIVATION

Main Idea



- Endow neural networks with **smoothness** through its **underlying function** from the perspective of **Partial Differential Equation**.
- Make the neural network from “only needs to fit the discrete training set” to “its function needs to satisfy the injective constraint throughout its entire domain”.

- Motivation
- **Method**
- Experiments

BACKGROUND

Partial Differential Equation (PDE)

- **What is ?** A kind of Equation contains unknown functions and their partial derivatives, whose solution is a function
- **What for?** Describing the relationship between the independent variables, unknown functions, and their partial derivatives.

$$\sum_i a_i \frac{\partial f}{\partial x_i} + bf = 0$$

$$\sum_{ij} a_{ij} \frac{\partial f}{\partial x_i \partial x_j} + \sum_i b_i \frac{\partial f}{\partial x_i} + cf = 0$$

BACKGROUND

Transport Equation (TE)

- **What is ?** A kind of PDE
- **What for ?** Describe the concentration of a quantity transport in a fluid.

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) + F(\mathbf{x}, \boldsymbol{\theta}(t)) \cdot \nabla u(\mathbf{x}, t) = 0$$

- $\mathbf{x}$: Variable
- t: Time
- u: Function of concentration
- F: Velocity field
- ∇: Gradient

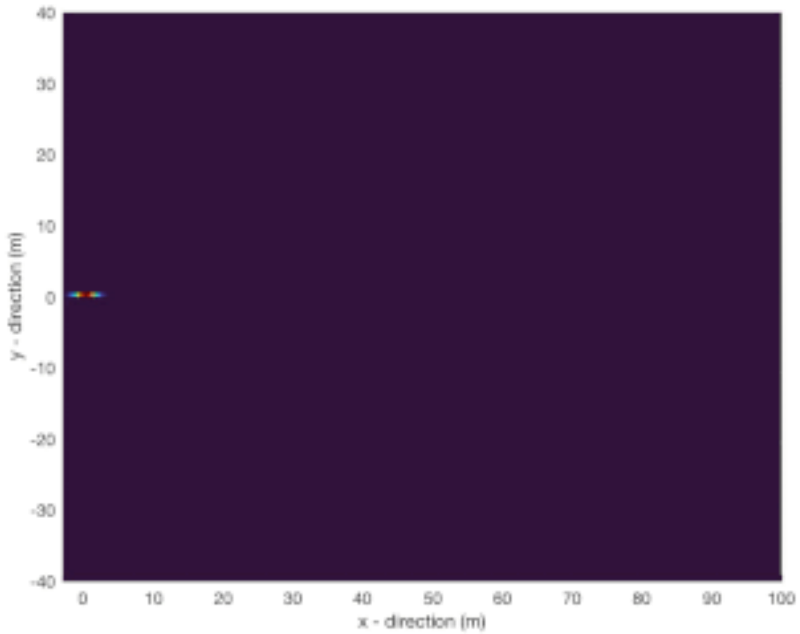


Figure: visualization of Transport Equation

METHOD: From PDE to NN

Transport Equation for Neural Network Modeling

- Change in particle position ⟺ Evolution of data representation

Table: The correspondence between Transport Equation and Neural Network.

Notations	Transport Equation	Neural Network
$u(\mathbf{x}, t)$	Function of concentration	Function of neural network
$t \in (0,1)$	Time	Layer
$\mathbf{x} \in \mathbb{R}^d$	Position of particles	Representation of samples
$F\big(x, \theta(t)\big)$	Velocity field	Continuation for network structures and parameters

METHOD: From PDE to NN

Solve Transport Equation for Network Function

Transport Equation

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) + F(\mathbf{x}, \boldsymbol{\theta}(t)) \cdot \nabla u(\mathbf{x}, t) = 0$$

⇓ Method of Characteristics

$$\mathrm{d}\mathbf{x}(t) = F(\mathbf{x}(t), \boldsymbol{\theta}(t))\, \mathrm{d}t$$

$$u(\hat{\mathbf{x}}, 0) = o\Big(\hat{\mathbf{x}} + \int_0^1 F(\mathbf{x}(t), \boldsymbol{\theta}(t))\mathrm{d}t\Big)$$

Euler Method ⇒

Residual Connection

$$\mathbf{h}_{l+1} = f(\mathbf{h}_l, \boldsymbol{\theta}_l) + \mathbf{h}_l$$

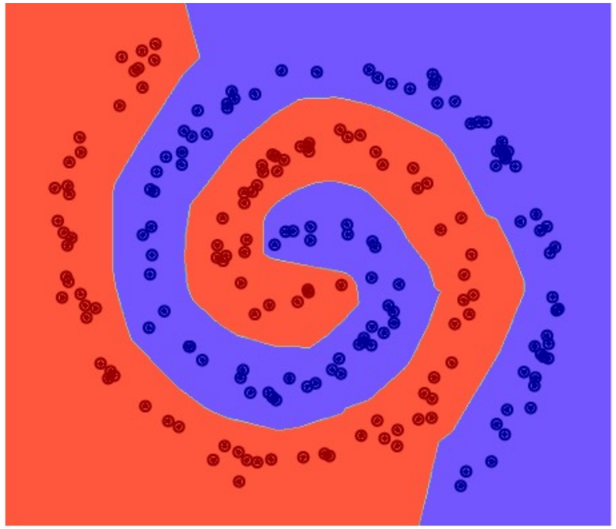
$$u(\hat{\mathbf{x}}, 0) = o\Big(\hat{\mathbf{x}} + \sum_{l=1}^{L} f(\mathbf{h}_l, \boldsymbol{\theta}_l)\Big)$$

METHOD: From PDE to NN

Smoothness boosts Generalization

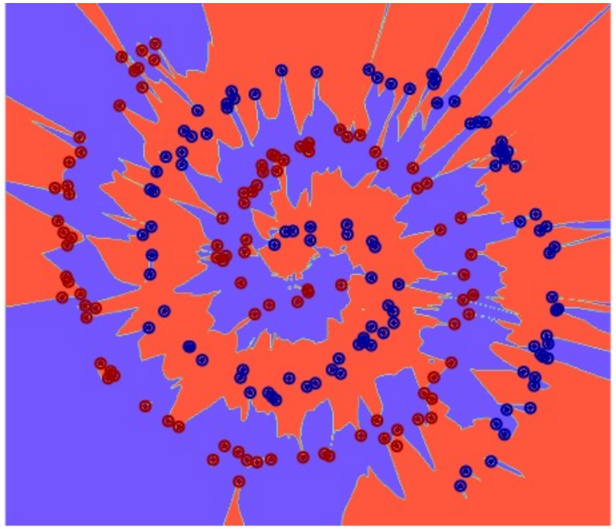
- Generalization is strongly linked to smoothness in neural networks.

Generalization of smooth classification hyperplane.



(a) 100% train, 100% test

Overfitting of non-smooth classification hyperplane.



(b) 100% train, 7% test

Image credit: Huang et al., Understanding Generalization Through Visualizations, 2020.

METHOD: From PDE to NN

Diffusion Term

- **What is?** Diffusion term $\Delta u(\mathbf{x}, t)$ corresponds to the Laplacian, i.e., the second-order derivative with respect to $\mathbf{x} \in \mathbb{R}^{\boldsymbol{d}}$.
- **What for?** Improve the smoothness of the PDE solution $u(\mathbf{x}, t)$.

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) + F(\mathbf{x}, \boldsymbol{\theta}(t)) \cdot \nabla u(\mathbf{x}, t) + \frac{1}{2}\sigma^2 \cdot \boxed{\Delta u(\mathbf{x}, t)} = 0$$

$$\Delta u = \partial^2 u / \partial x_1^2 + \partial^2 u / \partial x_2^2 + \cdots + \partial^2 u / \partial x_d^2$$

- Δ: Laplacian operator
- $\sigma \neq 0$: Coefficient for the diffusion magnitude

METHOD: From PDE to NN

Diffusion Term Brings Smoothness to TE

Theorem 1 (Proved in Appendix C.1) *Given TE with diffusion term (Eq. (6)) with terminal condition* $u(\mathbf{x}, 1) = o(\mathbf{x})$, *where* $F(\mathbf{x}, \boldsymbol{\theta}(t))$ *be a Lipschitz function in both* $\mathbf{x}$ *and* t, $o(\mathbf{x})$ *be a bounded function. Then, for any small* δ, $|u(\mathbf{x}+\delta, 0) - u(\mathbf{x}, 0)| \leq C\left(\frac{\|\delta\|_2}{\sigma}\right)^{\alpha}$ *holds for constant* $\alpha > 0$ *if* $\sigma \leq 1$, *where* $\|\delta\|_2$ *is the* ℓ_2 *norm of* δ, *and* C *is a constant that depends on* d, $\|o\|_\infty$, *and* $\|F\|_{L^\infty_{\mathbf{x},t}}$.

METHOD: From PDE to NN

Diffusion Term Brings Generalization to NN

Corollary 1 (Proved in Appendix C.2) *Generalization Error (GE) of model* $u(\mathbf{x}, 0)$ *trained on training set* s_N *is upper bounded by diffusion* σ*. For any* $\epsilon > 0$*, the following inequality holds with probability at least* $1 - \epsilon$*. For more details about the notations used, please refer to Appendix C.2.*

$$\mathrm{GE}\left(u(\mathbf{x}, 0), s_N\right) \leq C \cdot L \left(\frac{\|\delta'\|_2}{\sigma}\right)^{\alpha} + M \sqrt{\frac{2K \ln 2 + 2 \ln(1/\epsilon)}{N}}$$

METHOD: From PDE to NN

What type of diffusion term is appropriate for a neural network to achieve effective generalization?

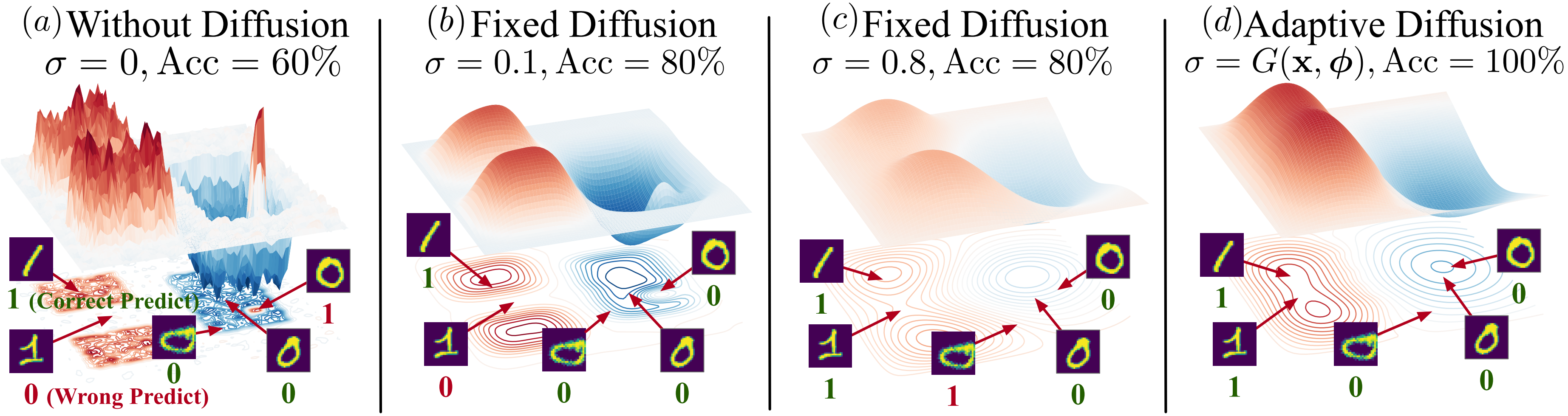


Figure: Solutions to 2D Transport Equation with different diffusion scales σ.

- Different locations $\mathbf{x}$ required **different** diffusion scales σ.
  - Uniform Small Diffusion: insufficient smoothness
  - Uniform Large Diffusion: over-smoothness.

METHOD: From PDE to NN

Adaptive Distributional Diffusion for Generalization

- **Adaptive Distributional Diffusion** is appropriate for a neural network to achieve effective generalization
  - **Adaptive**: diffusion varies in magnitude for every point.
  - **Distributional**: cover data spaces with similar semantics.
- Adaptive Distribution Diffusion (ADD)
  - $G(\mathbf{x}, \phi(t))$ parameterized diffusion term added to the Transport Equation.

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) + F(\mathbf{x}, \boldsymbol{\theta}(t)) \cdot \nabla u(\mathbf{x}, t) + \boxed{\frac{1}{2} G(\mathbf{x}, \phi(t))^2 \cdot \Delta u(\mathbf{x}, t)} = 0$$

METHOD: From PDE to NN

Deriving Neural Network from Transport Equation with Adaptive Distributional Diffusion

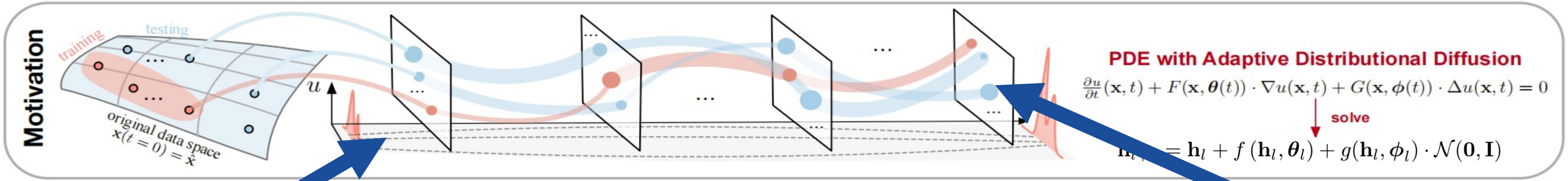
Theorem 2 (Proved in Appendix C.3) *TE with adaptive distributional diffusion term (Eq. (9))can be solved using the Feynman-Kac formula (Kac 1949), The result is shown in Eqs. (10) and (11), where B_t represents the Brownian motion (Uhlenbeck and Ornstein 1930).*

$$u(\hat{\mathbf{x}}, 0) = \mathbb{E}\left[o(\mathbf{x}(1)) \mid \mathbf{x}(0) = \hat{\mathbf{x}}\right] \tag{10}$$

$$\mathrm{d}\mathbf{x}(t) = F(\mathbf{x}(t), \boldsymbol{\theta}(t))\,\mathrm{d}t + G(\mathbf{x}(t), \boldsymbol{\phi}(t)) \cdot \mathrm{d}B_t \tag{11}$$

METHOD: From PDE to NN

Deriving Neural Network from Transport Equation with Adaptive Distributional Diffusion



Dynamical systems of PDEs = Feature transformation in NNs

$$\frac{\partial u}{\partial t}(\mathbf{x},t)+F(\mathbf{x},\boldsymbol{\theta}(t))\cdot\nabla u(\mathbf{x},t)+\boxed{\frac{1}{2}G(\mathbf{x},\phi(t))^2\cdot\Delta u(\mathbf{x},t)}=0$$

Induce ⇒

$$\mathbf{h}_{l+1}=\mathbf{h}_l+f(\mathbf{h}_l,\boldsymbol{\theta}_l)+g(\mathbf{h}_l,\phi_l)\cdot\mathcal{N}(\mathbf{0},\mathbf{I})$$

Adaptive **D**istributional **D**iffusion Term

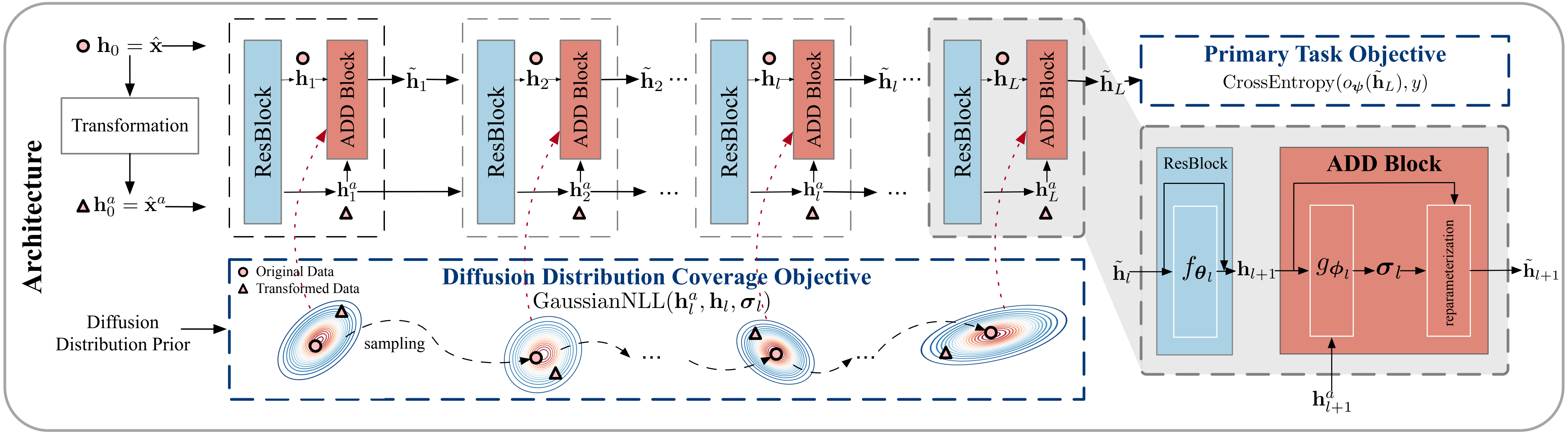
PDE: the solution is a function that satisfies the smoothness constraint terms injected into the equation.

Neural network architecture: Under this structure, any learned function satisfies desired smoothness.

METHOD: A Neural Network Instantiation

Architecture and Parameterization

- Overall Framework of PDE+:



$$\mathbf{h}_{l+1} = \mathbf{h}_l + f\left(\mathbf{h}_l, \boldsymbol{\theta}_l\right) \quad \boldsymbol{\sigma}_{l+1} = g_{\boldsymbol{\phi}_{l+1}}(\mathbf{h}_{l+1}) \quad \tilde{\mathbf{h}}_{l+1} = \mathbf{h}_{l+1} + \boldsymbol{\sigma}_{l+1} \cdot \mathcal{N}(\mathbf{0}, \mathbf{I})$$

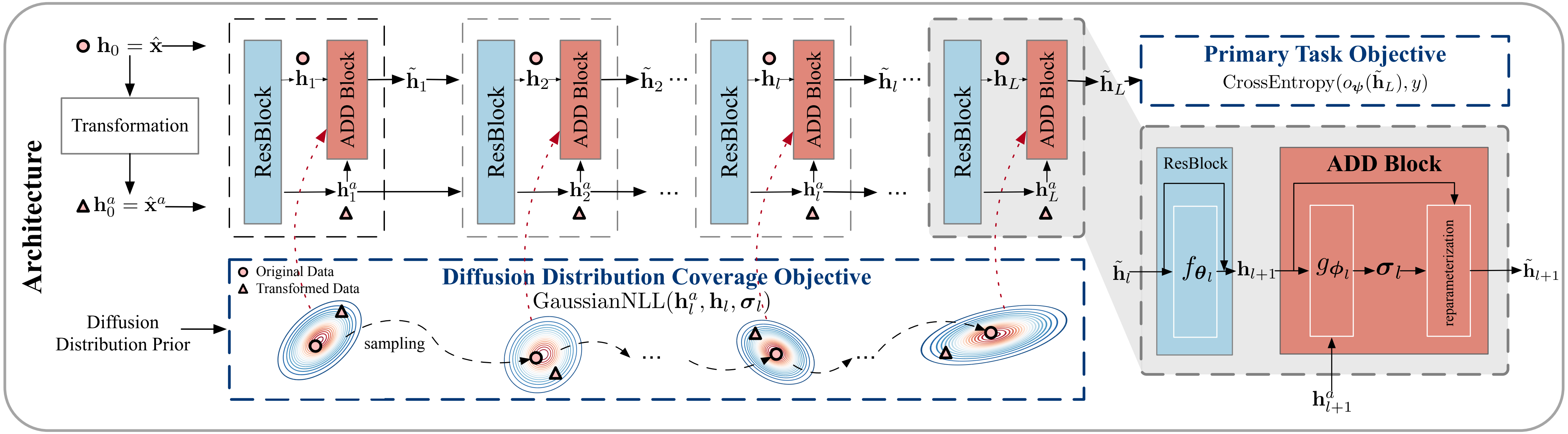
$$\text{PDE+}_{\boldsymbol{\theta},\boldsymbol{\phi}} : \left(g_{\boldsymbol{\phi}_l} \circ \left(f_{\boldsymbol{\theta}_{l-1}} + I\right) \circ \cdots \circ g_{\boldsymbol{\phi}_3} \circ \left(f_{\boldsymbol{\theta}_2} + I\right) \circ g_{\boldsymbol{\phi}_2} \circ \left(f_{\boldsymbol{\theta}_1} + I\right)\right)$$

METHOD: A Neural Network Instantiation

Learning Objectives

- Diffusion distributional coverage objective for the ADD block

$$\min_{\boldsymbol{\phi}} \mathbb{E}_{\mathbf{x}\sim s_N} - \sum_{l=1}^{L} \log p_{\boldsymbol{\phi}_l}(\mathbf{h}_l^a \mid \mathbf{h}_l) = -\frac{1}{2N}\sum_{n=1}^{N}\sum_{l=1}^{L}\left[\log g_{\boldsymbol{\phi}_l}(\mathbf{h}_l) + \frac{(\mathbf{h}_{n,l}^a - \mathbf{h}_{n,l})^2}{g_{\boldsymbol{\phi}_l}(\mathbf{h}_l)}\right]$$

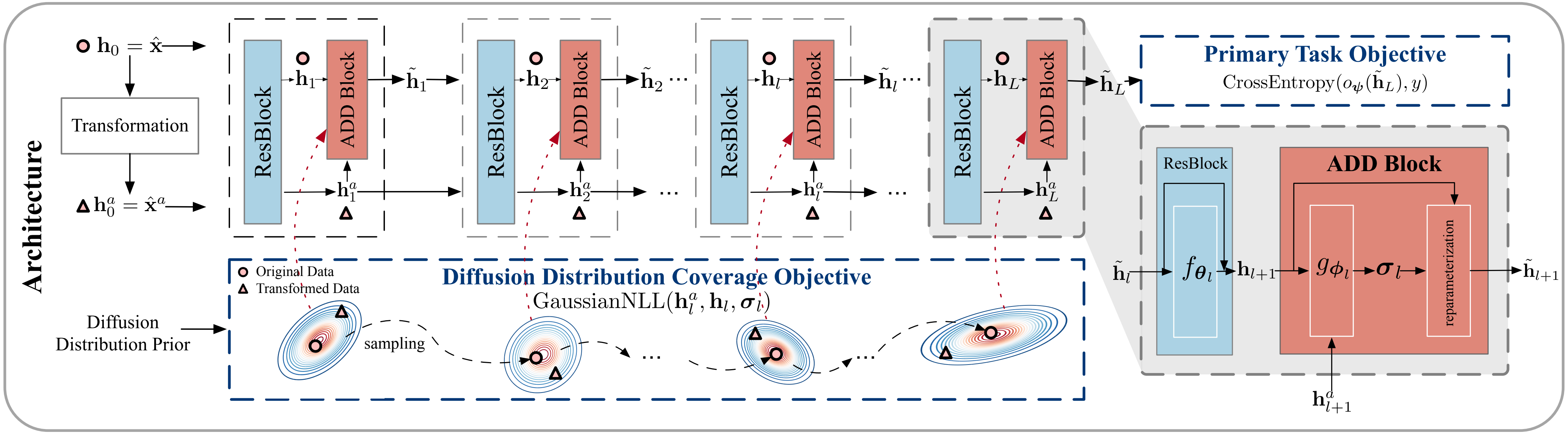


METHOD: A Neural Network Instantiation

Learning Objectives

- Primary task objective for the entire network

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}} \mathbb{E}_{(\mathbf{x},y)\sim s_N} -\log p_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}}(y \mid \mathbf{x}) = -\frac{1}{N}\sum_{n=1}^{N}\left[\log \frac{\exp(o_{\boldsymbol{\psi}}(\tilde{\mathbf{h}}_{n,L})_{y_n})}{\sum_{c=1}^{C}\exp(o_{\boldsymbol{\psi}}(\tilde{\mathbf{h}}_{n,L})_c)}\right]_{y_n}$$



- Motivation
- Method
- **Experiments**

EXPERIMENTS

Experimental Settings

- **Datasets**
  - 15 shift corruption distributions by **CIFAR-10(C), CIFAR-100(C), and Tiny-ImageNet(C)**.
  - 4 domains in the PCAS dataset: **photo, art, cartoon, and sketch**.
- **Baselines**
  - Standard training: **ERM**.
  - Lipschitz continuity based gradient regularization: **GradReg**.
  - Noise injection: **EnResNet, RSE, NFM**.
  - Data augmentation: **Gaussian noise, Mixup, DeepAug, AutoAug, AugMix**.
  - Adversarial training: **PGD, RLAT**.
- **Metrics**
  - **Accuracy** on clean data;**Accuracy** and **mCE** across all severity levels and at the severest level.

EXPERIMENTS

PDE+ Outperforms SOTA on Benchmarks

		CIFAR-10(C)					CIFAR-100(C)					Tiny-ImageNet(C)				
	Method	Clean	Corr Severity All		Corr Severity 5		Clean	Corr Severity All		Corr Severity 5		Clean	Corr Severity All		Corr Severity 5	
		Acc (↑)	Acc (↑)	mCE (↓)	Acc (↑)	mCE (↓)	Acc (↑)	Acc (↑)	mCE (↓)	Acc (↑)	mCE (↓)	Acc (↑)	Acc (↑)	mCE (↓)	Acc (↑)	mCE (↓)
Std	ERM	95.35	74.63	100.00	57.19	100.00	77.71	49.27	100.00	33.18	100.00	54.02	25.57	100.00	15.54	100.00
Lip	GradReg	93.64	77.62	96.29	62.33	91.52	73.80	52.16	96.95	37.33	94.49	52.01	29.20	95.13	19.91	94.86
NI	EnResNet	83.33	74.34	137.98	66.87	63.72	67.11	49.28	103.61	40.24	83.56	49.26	25.83	100.18	19.01	96.55
	RSE	95.59	77.86	94.12	63.66	89.08	77.98	53.73	94.10	38.03	92.88	53.74	27.99	96.81	18.92	96.11
	NFM*	95.40	83.30	-	-	-	79.40	59.70	-	-	-	-	-	-	-	-
DA	Gaussian	92.50	80.46	100.03	68.08	87.22	71.87	54.24	98.34	41.77	89.81	48.89	32.92	90.48	24.57	89.56
	Mixup*	**95.80**	80.40	-	-	-	**79.70**	54.20	-	-	-	-	-	-	-	-
	DeepAug*	94.10	85.33	64.63	77.29	60.05	-	-	-	-	-	**54.90**	-	-	-	-
	AutoAug	95.61	85.37	61.74	75.12	62.07	76.34	58.72	83.12	45.38	82.84	52.63	35.14	87.67	25.36	88.54
	AugMix	95.26	86.24	60.44	76.06	59.96	77.11	61.93	77.51	48.99	77.52	52.82	37.74	84.06	28.66	84.69
AT	PGD_{ℓ_∞}	93.52	82.17	86.53	70.10	78.20	71.78	55.03	93.49	42.04	88.17	49.94	32.54	90.65	23.47	90.63
	PGD_{ℓ_2}	93.91	83.07	81.06	70.97	75.17	72.50	56.09	91.65	42.82	87.33	51.08	33.46	89.37	24.00	89.92
	RLAT	93.23	83.67	80.98	72.73	72.59	71.10	56.54	91.98	44.27	86.24	50.24	33.13	89.83	24.46	89.47
	$RLAT_{Augmix}$	94.73	88.28	55.60	80.37	51.56	75.06	62.77	77.38	51.60	74.24	51.29	37.92	83.69	29.05	84.17
Ours	PDE+	95.59	**89.11**	**48.07**	**82.81**	**44.97**	78.84	**65.62**	**69.68**	**54.22**	**69.43**	53.72	**39.41**	**81.80**	**30.32**	**82.68**

EXPERIMENTS

PDE+ Outperforms SOTA on PACS

Source Domain	Method	Target Domain				Avg
		Photo	Art	Cartoon	Sketch	
Photo	ERM	-	21.33	22.31	28.35	24.00
	Augmix	-	26.90	24.10	27.05	26.02
	PDE+	-	**25.43**	**28.58**	**37.69**	**30.57**
Art	ERM	47.54	-	34.51	34.48	38.85
	Augmix	51.37	-	42.06	36.75	43.40
	PDE+	**53.11**	-	**43.90**	**41.28**	**46.10**
Cartoon	ERM	43.59	29.78	-	33.87	35.75
	Augmix	45.74	30.81	-	37.31	37.96
	PDE+	**48.68**	**33.00**	-	**40.01**	**40.57**
Sketch	ERM	18.74	16.16	25.26	-	20.05
	Augmix	26.28	26.51	45.34	-	32.72
	PDE+	**30.05**	**30.90**	**45.43**	-	**35.47**

Table: When training on a single source domain and testing on the remaining 3 domains, PDE+ surpasses the baselines across all splits.

EXPERIMENTS

PDE+ Learns Appropriate Diffusion

19 different test distributions on CIFAR-10-C

Standard	FixDiff 0.1	FixDiff 0.5	FixDiff 1.0	FixDiff 1.5	FixDiff 2.0	FixDiff 2.5	**PDE+ w/o aug**
45.47	45.87	51.66	48.12	51.43	51.89	47.51	60.98
52.12	53.10	56.29	57.35	58.42	58.19	55.32	61.72
50.51	50.01	55.56	56.27	57.19	58.24	54.60	62.99
49.05	50.51	53.40	53.69	48.78	47.13	48.15	59.17
54.34	54.49	57.67	58.44	56.62	55.29	53.22	63.86
50.50	52.22	53.47	54.65	51.17	44.00	44.86	56.36
51.58	52.32	55.34	55.91	51.48	48.98	48.57	60.76
48.36	49.43	52.65	53.14	47.71	45.53	44.57	59.25
57.08	57.04	60.38	61.14	60.05	59.37	57.19	62.82
56.72	56.95	60.30	61.32	59.90	59.12	57.48	62.70
51.74	52.41	55.18	55.88	53.47	50.89	50.26	54.38
55.24	55.84	58.72	60.14	56.41	55.07	54.88	58.79
59.08	58.82	62.62	62.70	65.02	65.85	61.36	67.80
58.32	57.96	61.41	62.29	62.66	59.67	58.05	66.09
54.53	55.02	58.63	59.80	62.46	64.23	60.05	63.80
52.58	53.03	55.80	57.19	57.32	57.32	54.10	63.35
56.91	56.22	60.81	62.01	65.81	66.82	62.86	69.15
57.58	55.46	60.98	60.25	64.95	64.69	57.90	65.84
37.50	36.51	42.41	35.82	38.86	34.77	33.35	45.72

EXPERIMENTS

PDE+ Generalizes Beyond Observation

2-σ rule

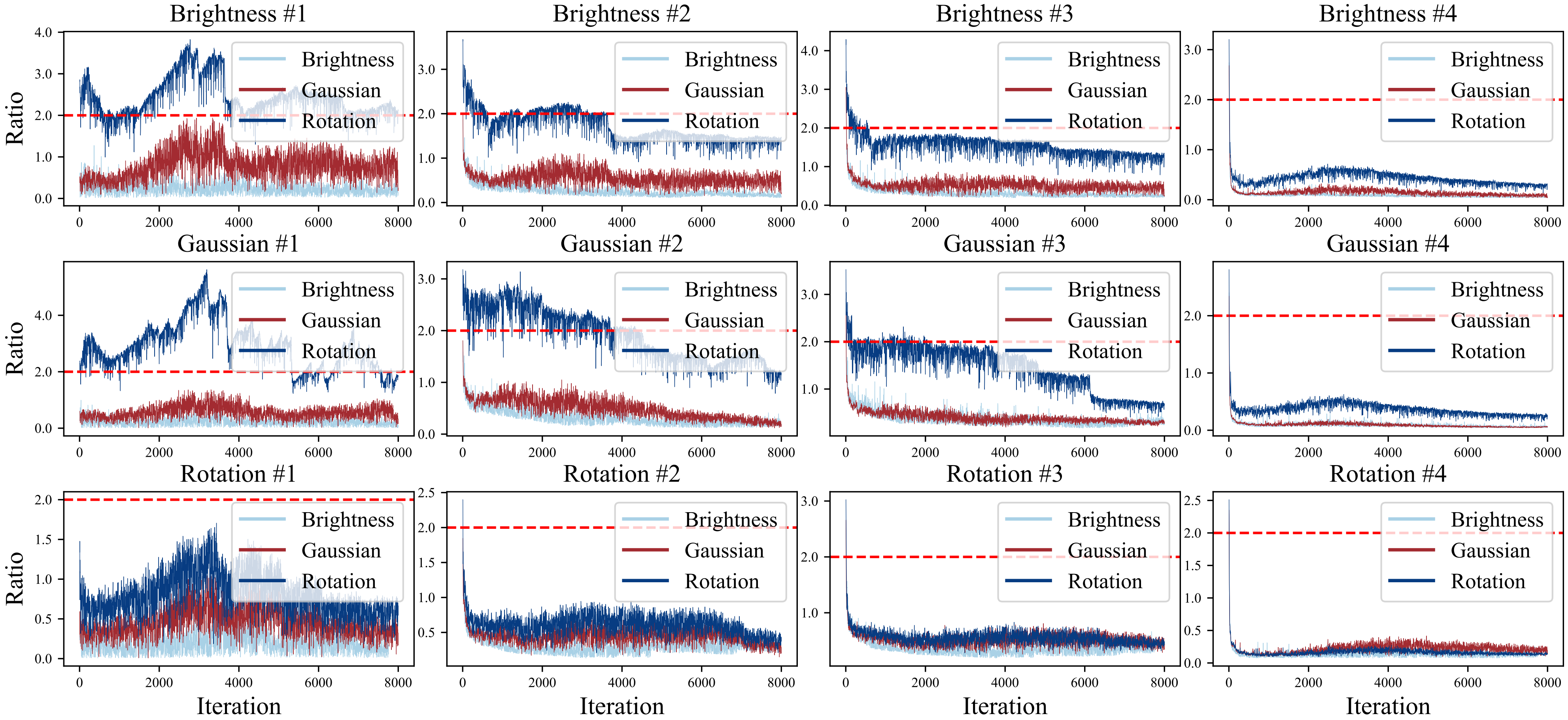


Figure: Diffusion coverage trend for unobserved distributions.

EXPERIMENTS

PDE+ Generalizes Beyond Observation

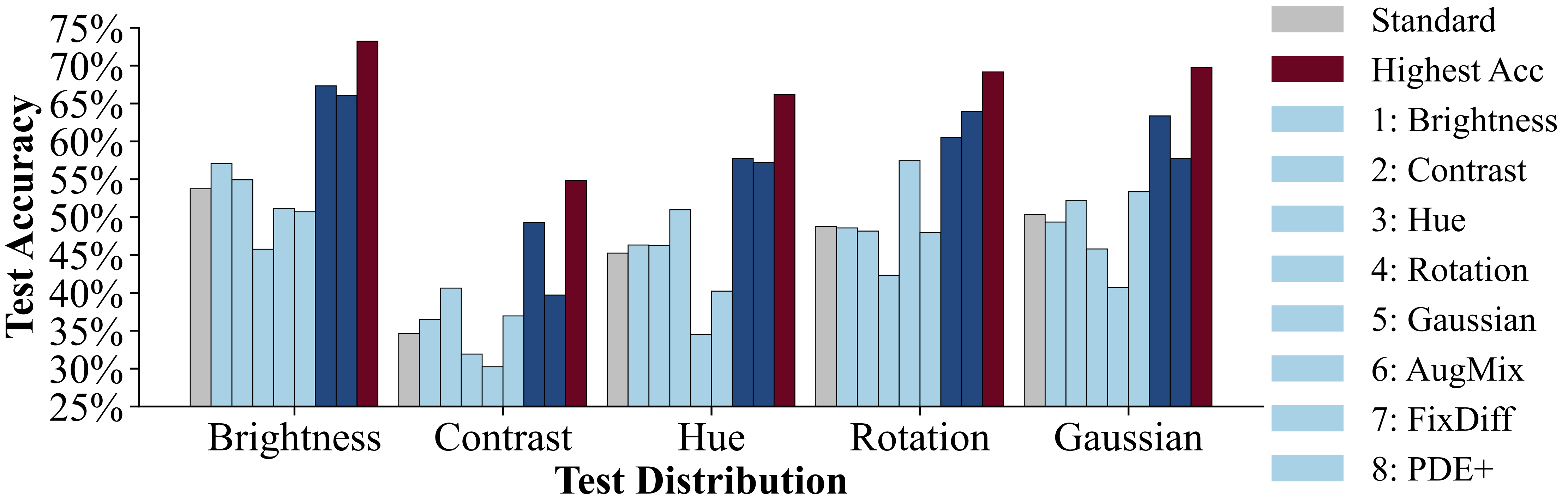


Figure: Generalization performance under five test distributions across eight different methods.

The 38th Annual AAAI Conference on Artificial Intelligence

FEBRUARY 20–27, 2024 | VANCOUVER, CANADA
VANCOUVER CONVENTION CENTRE – WEST BUILDING

Thanks for all the authors of this paper:

Yige Yuan
- Generalization
- Trustworthy AI

Bingbing Xu
- Graph Neural Networks
- Network Embedding

Bo Lin
- Numerical Methods
- Convection-Diffusion Equations

Liang Hou
- Generative Adversarial Nets
- Generative Models

Fei Sun
- Recommender Systems
- Natural Language Processing

Huawei Shen
- Network Data Mining
- Social Network Analysis
- Graph Neural Networks

Xueqi Cheng
- Network Data Science
- Social Computing
- Information Retrieval

Thank you for your attention!

Paper	Code	Homepage	WeChat

Email: yuanyige20z@ict.ac.cn