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 & Security,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Department of Mathematics, National University of Singapore, Singapore
{yuanyige20z, xubingbing, houliang17z, sunfei, shenhuawei, cxq}@ict.ac.cn, matbl@nus.edu.sg

Abstract

The generalization of neural networks is a central challenge in machine learning, especially concerning the performance under distributions that differ from training ones. Current methods, mainly based on the data-driven paradigm such as data augmentation, adversarial training, and noise injection, may encounter limited generalization due to model non-smoothness. In this paper, we propose to investigate generalization from a Partial Differential Equation (PDE) perspective, aiming to enhance it directly through the underlying function of neural networks, rather than focusing on adjusting input data. Specifically, we first establish the connection between neural network generalization and the smoothness of the solution to a specific PDE, namely transport equation. Building upon this, we propose a general framework that introduces adaptive distributional diffusion into transport equation to enhance the smoothness of its solution, thereby improving generalization. In the context of neural networks, we put this theoretical framework into practice as **PDE+** (**PDE** with **A**daptive **D**istributional **D**iffusion) which diffuses each sample into a distribution covering semantically similar inputs. This enables better coverage of potentially unobserved distributions in training, thus improving generalization beyond merely data-driven methods. The effectiveness of PDE+ is validated through extensive experimental settings, demonstrating its superior performance compared to state-of-the-art methods. Our code is available at https://github.com/yuanyige/pde-add.

1 Introduction

The generalization of neural networks is a fundamental challenge in the field of machine learning. It refers to the ability of neural networks to perform effectively under unobserved distributions, which may differ from those encountered during the training process (Bousquet and Elisseeff 2002). Pursuing superior generalization capability is essential as it ensures model adaptability to diverse real-world scenarios, guaranteeing reliable predictions and decisions.

Existing approaches for improving generalization mainly employ a data-driven paradigm (Emmert-Streib and Dehmer 2022), including data augmentation (Shorten and Khoshgoftaar 2019), adversarial training (Madry et al. 2018), and noise injection (Bishop 1995). In terms of implementation, they primarily enhance the training samples via manipulating the original input (Hendrycks et al. 2020; Madry et al. 2018) or transforming the hidden representations (Lim et al. 2022). However, such a data-driven paradigm usually cannot guarantee reliable generalization capabilities on unobserved distributions. Taking data augmentation as an illustration, Fig. 1 shows that the model can only achieve satisfactory generalization performance when the training data is subjected to augmentation similar to that of the testing data. Analogous phenomena are also frequently observed in adversarial training and noise injection. For instance, adversarial training can improve generalization on adversarial examples but often comes with the cost of performance on natural data (Tsipras et al. 2019). Likewise, while injecting Gaussian noise can enhance generalization in the face of common corruptions, it risks σ-overfitting (Klim, Maksym, and Nicolas 2022), i.e., overfitting to the particular Gaussian noise used in training.

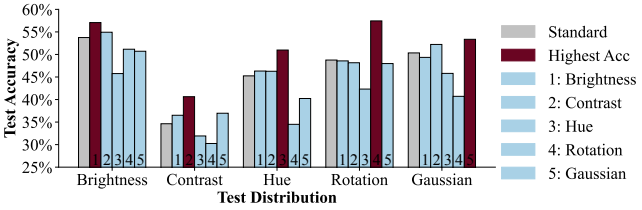

Figure 1: Model trained on six training distributions and evaluated on five corresponding test distributions. Model performs best on each test distribution is highlighted in red.

The limited generalization capabilities of data-driven paradigm is due to model *irregularity* (Wang et al. 2020), i.e., the function learned by the neural network is non-smoothness. This may cause a problematic situation where semantically similar samples are encoded distantly, resulting in incorrect predictions. To address the irregularity issue, several approaches have been proposed to improve the smoothness of models (Sokolic et al. 2017), which helps to tackle the distribution shift problem (Rodríguez et al. 2020). Among them, Lipschitz continuity (Cisse et al. 2017) enforces smoothness constraints on models through regularization or architectural restrictions, e.g., gradient regularization (Drucker and Le Cun 1992) and spectral normalization (Miyato et al. 2018). However, such restrictions often come at the cost of expressive power (Anil, Lucas, and Grosse 2019).

*Corresponding author

In this paper, we go beyond the data-driven paradigm and propose to investigate generalization from a Partial Differential Equation (PDE) (Arrigo 2017) perspective, aiming to directly introduce the smoothness constraint into the underlying function fn of neural network, rather than manipulating input data. The feasibility of this perspective is rooted in the intrinsic connection between neural networks and PDE (E 2017). PDE describes a function that satisfies differential relationships, and neural networks can be regarded as a discrete numerical difference solver of PDE. That is to say, the underlying function of a neural network can be considered as the solution to PDE (Li and Shi 2017; Han, Jentzen, and E 2018). From such perspective, we can leverage the vast prior knowledge of PDE to constrain the underlying function of neural network, thus encouraging the resulting neural networks to exhibit specific desired properties, e.g., smoothness (Wang et al. 2020), well-posedness (Haber and Ruthotto 2017), and hyperbolicity (Eliasof, Haber, and Treister 2021).

The above fundamental connection inspires us to establish the connection between neural network generalization and the smoothness of PDE solution. Specifically, we initially model the neural network as the solution of a specific type of PDE, referred to as transport equation (TE) (Li and Shi 2017), which is often employed to describe the transportation of a quantity within a space. Then, a diffusion term is introduced into the TE, which has been proven to smooth the solution (Ladyzhenskaia, Solonnikov, and Ural'tseva 1968). The core of such paradigm is this key question: *What type of diffusion term is appropriate for a neural network to achieve effective generalization?* To answer it, we propose a general framework that introduces adaptive distributional diffusion into transport equation to enhance the smoothness of its solution. Such diffusion ensures suitable smoothness by treating the diffusion scope of each sample as a distribution that should cover the potential semantically similar inputs, thus improving generalization.

In the context of neural networks, we put this theoretical framework into practice as **PDE+** (**PDE** with **A**daptive **D**istributional **D**iffusion, PDE-ADD) to achieve generalization. Specifically, we introduce adaptive distributional diffusion into the neural network, which performs diffusion centered on each data point. The scope of each diffusion is modeled as a distribution, determined adaptively by multiple augmentations of the input. This enables better coverage of potentially unobserved distributions and improves generalization beyond data-driven approaches. The effectiveness of PDE+ is validated on various distributions, including clean samples and various common corruptions. The consistent improvements demonstrate the superior performance of our method over state-of-the-art methods.

Our main contributions include:

(1) *A promising paradigm*: we investigate generalization from a Partial Differential Equation (PDE) perspective. To the best of our knowledge, we are the first to achieve generalization by establishing connections between the generalization of neural networks and the smoothness of TE solutions.

(2) *An innovative method*: we propose an adaptive distributional diffusion term to incorporate smoothness into a neural network and instantiate it as PDE+, enabling better coverage of potentially unobserved distributions in training and improves generalization compared to data-driven methods.

(3) *Solid experiments*: extensive experiments reveal PDE+ outperforms baselines across unobserved distributions, e.g., the improvements are up to 3.8% in Acc and 7.7% in mCE.

2 Related Work

In this section, we briefly review two lines of research that close to our work: the generalization of neural networks and differential equations based neural networks. Detailed introduction of related works can be found in Appendix B[1].

Generalization of Neural Networks. Current data-driven methods encompass data augmentation, adversarial training, and noise injection. Data augmentation is a widely adopted technique to enhance generalization, employing various strategies such as Mixup (Zhang et al. 2018)and AugMix (Hendrycks et al. 2020). Adversarial training is a robust optimization approach for improving adversarial generalization (Goodfellow, Shlens, and Szegedy 2015) while potentially compromising non-adversarial generalization (Tsipras et al. 2019; Zhang et al. 2019). Notable works in this area include PGD (Madry et al. 2018), TRADES (Zhang et al. 2019), and RLAT (Klim, Maksym, and Nicolas 2022). Noise injection introduces noise into input data (An 1996), activations (Gulcehre et al. 2016), or hidden layers (Camuto et al. 2020), whose noise magnitude can be sensitive and susceptible to overfitting (Klim, Maksym, and Nicolas 2022). Lipschitz continuity is often used to ensure model generalization (Drucker and Le Cun 1992; Miyato et al. 2018; Liu et al. 2023), but its strict constraint can restrict a model's capabilities (Anil, Lucas, and Grosse 2019). Our method diverges from above approaches, as we directly constrain the smoothness of the neural network's underlying function rather than fitting a finite set of input data like data-driven methods. Although our method shares the concept of smoothness with Lipschitz, it avoids compromising the model's capabilities.

Differential Equations based Neural Networks The connection between continuous dynamical systems and residual neural networks (He et al. 2016) is initially established in (E 2017). Subsequently, numerous studies have delved into the relationships between various neural network architectures and different types of differential equations (Lu et al. 2018; Li and Shi 2017; Sun, Tao, and Du 2018). Since then, researchers have started to explore the beneficial properties of differential equations to enhance neural networks (Wang and Lin 1998; Li, He, and Lin 2020; Wang et al. 2020).

3 Generalization under PDEs with Adaptive Distributional Diffusion

This section introduces the theoretical motivation and framework behind our method. We begin by establishing connections between PDEs and neural networks, thereby transforming the generalization of neural networks into the smoothness of PDE solutions. Our innovative adaptive distributional diffusion term is then introduced to enhance the smoothness of solutions, which improves generalizability.

[1]Appendix can be found in this version (Yuan et al. 2023)

3.1 Neural Network as the Solution of Transport Equation

Partial Differential Equation (PDE) (Arrigo 2017) is an equation containing an unknown function u of multiple variables and its partial derivatives. The connection between PDEs and neural networks has been discussed in (E 2017), where neural networks could be interpreted as a numerical scheme to solve PDEs. Such connection allows us to take advantage of PDE, such as the properties of solution as well as the numerical schemes, to obtain a better neural network. In this section, we make use of the transport equation (TE), which is one special form of PDE, to interpret neural networks.

TE describes the concentration of a quantity transport in a fluid (Pogodaev 2016; Munson et al. 2006) (Eq. (1)), which is suitable to model the feature transformation of data flow. This observation has also been discussed in (Li and Shi 2017; Sun, Tao, and Du 2018)

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) + F(\mathbf{x}, \boldsymbol{\theta}(t)) \cdot \nabla u(\mathbf{x}, t) = 0 \tag{1}$$

where $u(\mathbf{x}, t)$ denotes a function of concentration, which can be viewed as the underlying function of a neural network. $t \in (0, 1)$ denotes time, serving as the continuation of network layers. $\mathbf{x} \in \mathbb{R}^d$ denotes a variable in space, serving as the variable for data representation in terms of neural networks. ∇ represents gradient, and $F(\mathbf{x}, \boldsymbol{\theta}(t))$ is the velocity field, serving as the continuation for network structures and parameters. In terms of neural networks, the changing of representation through layers can be viewed as a transport process over time. The representation is transported through each layer, where the parameters of each layer serve as a velocity field aiming to make changes to the sample representations and transport it to the next layer. Given the parameters of all layers, the representation transforms from the original input to final output, acting like a transport of data flow as illustrated in the top subfigure of Fig. 3.

$u(\mathbf{x}, t)$ represents the value obtained by transporting the variable $\mathbf{x}$ through a series of $F(\mathbf{x}, \boldsymbol{\theta}(t))$ from time t until the terminal. The terminal condition of TE is enforced at $t = 1$ as $u(\mathbf{x}, 1) = o(\mathbf{x})$, where $o(\mathbf{x})$ denotes the output function such as softmax (Gold, Rangarajan et al. 1996). Let $\hat{\mathbf{x}}$ denote the input feature. The original data-label pair $(\hat{\mathbf{x}}, y)$ is given at $t = 0$, and an optimal network u^* should exactly maps $\hat{\mathbf{x}}$ to y, i.e., $u^*(\hat{\mathbf{x}}, 0) = y$. Obtaining the network is equivalent to solving the numerical solution of TE at $t = 0$ as $u(\hat{\mathbf{x}}, 0)$, where the method of characteristics (Sarra 2003) can be effectively employed. The main idea of the characteristics is to solve PDE via an ordinary differential equation (ODE) defining the characteristic curves of original PDE, which is shown in Eq. (2). Then the solution of PDE can be acquired by following these curves in Eq. (3).

$$\mathrm{d}\mathbf{x}(t) = F(\mathbf{x}(t), \boldsymbol{\theta}(t))\,\mathrm{d}t \tag{2}$$

$$u(\hat{\mathbf{x}}, 0) = o\Big(\hat{\mathbf{x}} + \int_0^1 F(\mathbf{x}(t), \boldsymbol{\theta}(t))\mathrm{d}t\Big) \tag{3}$$

To solve Eq. (2) numerically, we adopt Euler method (Butcher 2003, Chapter 2) as shown in Eq. (4), which recovers the formulation of ResNet (He et al. 2016). $l \in \{1, \ldots, L\}$ is the network layer index, serving as a discrete slicing to continuous time t. $\mathbf{h}_l$ and $\boldsymbol{\theta}_l$ are representations and parameters at layer l, respectively.

$$\mathbf{h}_{l+1} = f(\mathbf{h}_l, \boldsymbol{\theta}_l) + \mathbf{h}_l \tag{4}$$

$$u(\hat{\mathbf{x}}, 0) = o\Big(\hat{\mathbf{x}} + \sum_{l=1}^{L} f(\mathbf{h}_l, \boldsymbol{\theta}_l)\Big) \tag{5}$$

Overall, neural network, particularly ResNet can be seen as a solution to TE. This connection lays a solid foundation to achieve desired properties of neural networks by constraining the solution of TE.

3.2 Improving Generalization via Enhancing the Smoothness of TE Solution

Smoothness has been demonstrated to be strongly linked to generalization, as it facilitates models to generalize beyond the training distribution (Rosca et al. 2020; Rodríguez et al. 2020), enhances model robustness against small perturbations (Cisse et al. 2017; Sokolic et al. 2017), and plays a significant role in generalization quantization (Jin et al. 2020; Ng et al. 2022) as well as uncertainty estimation (Van Amersfoort et al. 2020; Liu et al. 2020). Building upon the insights, we propose to achieve generalization from the perspective of PDEs by modeling neural networks as solutions to PDEs and transforming the generalization goal of neural networks into smoothness goal of a solution to PDEs.

To enhance the smoothness of solution $u(\mathbf{x}, t)$, we leverage knowledge from PDE field to introduce a diffusion term (Ladyzhenskaia, Solonnikov, and Ural'tseva 1968) $\Delta u(\mathbf{x}, t)$ into TE as Eq. (6). The diffusion term corresponds to the Laplacian, i.e., the second-order derivative with respect to $\mathbf{x} \in \mathbb{R}^d$, as illustrated in Eq. (7). Here, Δ denotes the Laplacian operator, and $\sigma \neq 0$ is a coefficient for the diffusive magnitude.

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) + F(\mathbf{x}, \boldsymbol{\theta}(t)) \cdot \nabla u(\mathbf{x}, t) + \frac{1}{2}\sigma^2 \cdot \Delta u(\mathbf{x}, t) = 0 \tag{6}$$

$$\Delta u = \partial^2 u / \partial x_1^2 + \partial^2 u / \partial x_2^2 + \cdots + \partial^2 u / \partial x_d^2 \tag{7}$$

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, $|u(\mathbf{x}+\delta, 0) - u(\mathbf{x}, 0)| \leq C\big(\frac{\|\delta\|_2}{\sigma}\big)^{\alpha}$ 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 $\|F\|_{L^\infty_{\mathbf{x},t}}$.*

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.*

$$\mathrm{GE}\left(u(\mathbf{x}, 0), s_N\right) \leq C \cdot L \left(\frac{\|\delta'\|_2}{\sigma}\right)^{\alpha} + M\sqrt{\frac{2K \ln 2 + 2\ln(1/\epsilon)}{N}} \tag{8}$$

Typically, σ is chosen as a fixed scalar, imposing an uniform diffusion scale across entire data space (Wang et al. 2020). Fixed diffusion brings smoothness into TE solution,

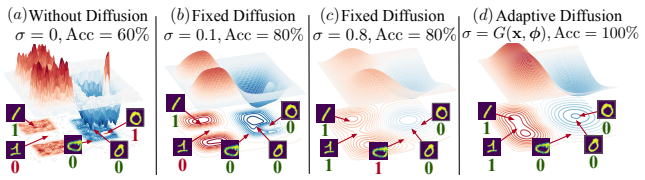

Figure 2: Solutions to 2D TE differs in the diffusion σ. The upper displays function surface, the lower exhibits its contour, with samples showing its true and predicted label.

but it neglects structure of solution for different $\mathbf{x}$. It cannot achieve an optimal diffusion scale for network across data space, as different locations require diverse diffusion scales based on their distance to other samples or class boundaries. To intuitively introduce the influence of diffusion, we illustrate the solution surface of 2D transport equation under different diffusion terms in Fig. 2. No diffusion in (a) results in highly irregular surface. Fixed diffusion with a small coefficient in (b) imposes insufficient smoothness for same class. Larger coefficient in (c) imposes over-smoothness for different classes. It reveals that a fixed coefficient can result in over-smoothness which diminishes variability, or insufficient smoothing. Thus, a new diffusion term is required to improve generalizability.

3.3 Adaptive Distributional Diffusion for Generalization

With concerns draw above, a crucial question arises:

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

To address it, we claim that a good diffusion term for generalization should satisfy two goals: "Adaptive" and "Distributional". "Adaptive" stands for that the diffusion varies in magnitude for every point across the entire space. "Distributional" treats the diffusion scope of each point as a distribution. For any input from the data space at any time step, the distribution should only encompass the inputs that are potentially similar to the central point in semantics. This mechanism allows for better coverage of potential unseen distributions and improved generalization compared to data-driven methods.

To achieve the above goals, we propose an **Adaptive Distributional Diffusion (ADD)** term and introduce it into TE as presented in Eq. (9). Rather than using a fixed scalar, our term incorporates a coefficient function $G(\mathbf{x}, \boldsymbol{\phi}(t))$ that takes sample $\mathbf{x}$ as input and outputs its diffusion scale, exhibiting different diffusion properties, based on the parameters $\boldsymbol{\phi}$ at each time step t. The benefits of the term can be illustrated in Fig. 2(d), which allows for different smoothing effects across space in accordance with the principle of "adaptive". Meanwhile, data spaces with similar semantics or within the same class can achieve smoothness in their scope, and those within different classes can avoid over-smoothness and maintain discrepancy. These satisfy the principle of "distributional".

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) + F(\mathbf{x}, \boldsymbol{\theta}(t)) \cdot \nabla u(\mathbf{x}, t) + \frac{1}{2} G(\mathbf{x}, \boldsymbol{\phi}(t))^2 \cdot \Delta u(\mathbf{x}, t) = 0 \tag{9}$$

3.4 Deriving Neural Network from Transport Equation with ADD

Introducing adaptive distributional diffusion into TE as Eq. (9) can realize the smoothness of the solution of TE, and thus encourage the resulting neural networks to exhibit generalization. In the following, we solve TE with ADD (Eq. (9)) to derive its corresponding neural network.

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).*

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

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

The result is a conditional expectation with respect to the initial value problem of stochastic differential equation (SDE, (Kloeden et al. 1992)) in Eq. (11). To obtain the final functional form of our neural network, we adopt the Euler–Maruyama method (Gelbrich and Römisch 1995) to compute the solution of SDE numerically as follows.

$$\begin{aligned} u(\hat{\mathbf{x}}, 0) &= \mathbb{E}\left[o(\mathbf{h}_L) \mid \mathbf{h}_0 = \hat{\mathbf{x}}\right] \\ \mathbf{h}_{l+1} &= \mathbf{h}_l + f\left(\mathbf{h}_l, \boldsymbol{\theta}_l\right) + g(\mathbf{h}_l, \boldsymbol{\phi}_l) \cdot \mathcal{N}(\mathbf{0}, \mathbf{I}) \end{aligned} \tag{12}$$

4 PDE+ : An Neural Network Instantiation

This section is for the instantiation of our framework **PDE+**: **PDE** with **A**daptive **D**istributional **D**iffusion (PDE-ADD).

4.1 Overall Architecture

PDE+ is a neural network instantiation of PDE solution formulated in Eq. (12), where $\mathbf{h}_{l+1} = \mathbf{h}_l + f\left(\mathbf{h}_l, \boldsymbol{\theta}_l\right)$ is the formulation for residual block, and $g(\mathbf{h}_l, \boldsymbol{\phi}_l) \cdot \mathcal{N}(\mathbf{0}, \mathbf{I})$ is implemented as our adaptive distributional diffusion block, dubbed as ADD block. As shown in Fig. 3, the residual block is denoted as $f_{\boldsymbol{\theta}_l}$ parameterized by $\boldsymbol{\theta}_l$, where $l \in \{1, \ldots, L\}$ denotes the block index. ADD block is denoted as $g_{\boldsymbol{\phi}_l}$ parameterized by $\boldsymbol{\phi}_l$. The overall architecture of PDE+, denoted as $fn_{\boldsymbol{\theta}, \boldsymbol{\phi}}$ is the composition of L blocks, where each block contains a residual block followed by our ADD block.

4.2 Adaptive Distributional Diffusion Block

Fig. 3 illustrates the structure of ADD block, which takes the output from residual block $\mathbf{h}_l$ as input, and outputs the scale $\boldsymbol{\sigma}_l$ for diffusion (Eq. (13)). Then a reparameterization trick (Kingma and Welling 2013) of $\mathbf{h}_l$ and $\boldsymbol{\sigma}_l$ under the prior of Gaussian distribution is conducted to obtain the final output $\tilde{\mathbf{h}}_l$ (Eq. (14)).

$$\boldsymbol{\sigma}_l = g_{\boldsymbol{\phi}_l}(\mathbf{h}_l) \tag{13}$$

$$\tilde{\mathbf{h}}_l = \mathbf{h}_l + \boldsymbol{\sigma}_l \cdot \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{14}$$

As introduced in Section 3.3, the principle of ADD blocks is "adaptive" and "distributional". "Adaptive" is implemented by replacing the fixed diffusion with the learnable $\boldsymbol{\sigma}_l$. "Distributional" means that for any input from the data space at any given time step, the diffusion scope should encompass the potential neighbors that exhibit semantic similarity. To achieve

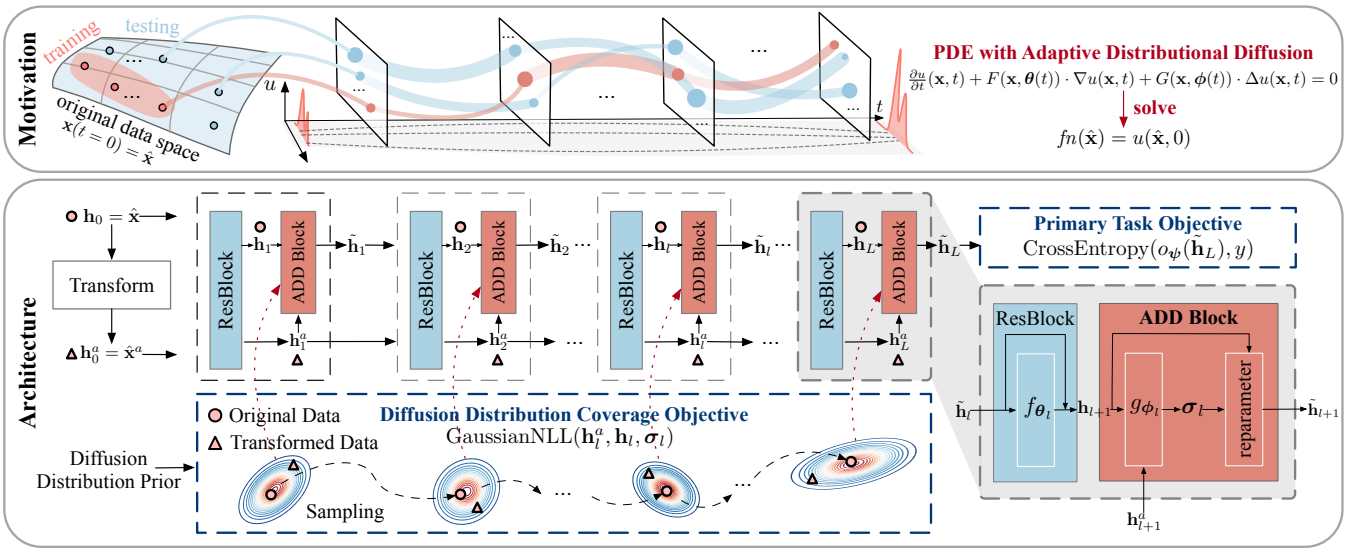

Figure 3: Motivation and architecture of PDE+, the upper illustrates our motivation of solving transport equation with adaptive distributional diffusion (ADD) to derive the functional form of neural network. The lower is the neural network instantiation, which comprises a series of blocks that contain a residual block followed by an ADD block. The architecture of the ADD block is enclosed grey frame on the right. The learning objectives are enclosed in two blue frames.

this, semantically similar samples are utilized as guidance. Define training dataset s_N containing N training samples of C classes $s_N = \{(\mathbf{x}_n, y_n) \mid n \in 1, 2 \dots N\}$. Let $\mathbf{x}_n^a$ represent samples that share semantic similarity with $\mathbf{x}_n$, such as augmented samples, style-transferred samples, or adversarial attack samples. We hope the diffusion distribution scope of $\mathbf{x}_n$ can cover $\mathbf{x}_n^a$.

To achieve this, we let the original samples pass through the whole block with both residual block and ADD block, and the semantically similar samples only go through residual block without diffusion. Denote I as the identity function where $I(x) = x$. The l-th layer's representation of original samples and their semantically similar counterparts can be formulated as... Eqs. (15) and (16), .

$$\tilde{\mathbf{h}}_l = (g_{\boldsymbol{\phi}_l} \circ (f_{\boldsymbol{\theta}_{l-1}} + I) \circ \cdots \circ g_{\boldsymbol{\phi}_2} \circ (f_{\boldsymbol{\theta}_1} + I))(\mathbf{x}) \quad (15)$$

$$\mathbf{h}_l^a = ((f_{\boldsymbol{\theta}_{l-1}} + I) \circ \cdots \circ (f_{\boldsymbol{\theta}_1} + I))(\mathbf{x}^{\mathbf{a}}) \quad (16)$$

For every block, the diffused hidden representation $\tilde{\mathbf{h}}_l$ can be regarded as a sampling from a Gaussian distribution $\mathcal{N}(\mathbf{h}_l, \boldsymbol{\sigma}_l)$, where the representations of semantically similar samples $\mathbf{h}_l^a$ should be covered. This objective can be implement via maximizing the probability of $\mathbf{h}_l^a$ under $\mathcal{N}(\mathbf{h}_l, \boldsymbol{\sigma}_l)$ denoted as $p_{\boldsymbol{\phi}}(\mathbf{h}_l^a \mid \mathbf{h}_l)$, which is equivalent to minimizing its negative log-likelihood. We named such objective as *diffusion distribution coverage objective* shown in Eq. (17), guiding only the parameters of diffusion blocks $\boldsymbol{\phi}$.

$$\begin{aligned} &\min_{\boldsymbol{\phi}} \mathop{\mathbb{E}}_{\mathbf{x}\sim s_N} -\sum_{l=1}^{L} \log p_{\boldsymbol{\phi}_l}(\mathbf{h}_l^a \mid \mathbf{h}_l) \\ &= -\frac{1}{2N}\sum_{n=1}^{N}\sum_{l=1}^{L}\left[\log g_{\boldsymbol{\phi}_l}(\mathbf{h}_l) + \frac{(\mathbf{h}_{n,l}^a - \mathbf{h}_{n,l})^2}{g_{\boldsymbol{\phi}_l}(\mathbf{h}_l)}\right] \end{aligned} \quad (17)$$

From a distributional point of view, the intuitive interpretation of our adaptive distributional diffusion is treating each sample as one distribution whose scope includes its semantic similar samples. Under such view, the basic residual block without diffusion treats each sample as a Dirac distribution (Cohen and Kirschner 1991) and our ADD block transforms it into Gaussian distribution. To broaden the distribution and enhance generalization, we advance from a single Gaussian to a Gaussian mixture (Reynolds et al. 2009), as it is a universal approximator of densities (Goodfellow, Bengio, and Courville 2016). Notably , we do not model the Gaussian mixture distribution directly. Rather, we allow both the original sample and its augmentations to diffuse simultaneously, effectively acting as different Gaussian centers. As a result, the superimposition of these multiple single Gaussians manifests as a mixed Gaussian from a macroscopic perspective. This implementation can be easily achieved in one line of code, as shown in Algorithm 1(Line 12) from Appendix D.

4.3 Learning Objectives

PDE+ consists of two learning objectives: a diffusion distribution coverage objective for every ADD block (Eq. (17)) and a primary task objective for the entire network. The primary task objective ensures the correctness of learning representations under diffusion. Define the output of $\mathbf{x}_n$ throughout the whole model fn as $\tilde{\mathbf{h}}_{n,L} = fn_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}_n)$. The primary task objective is shown in Eq. (18), where $o_{\boldsymbol{\psi}}$ stands for output layer parameterized by $\boldsymbol{\psi}$. The samples diffused throughout fn to obtain a classification probability via softmax, guiding the learning of all parameters, including residual blocks $\boldsymbol{\theta}$, diffusion blocks $\boldsymbol{\phi}$ and output $\boldsymbol{\psi}$ via cross-entropy. The algorithmic pseudocode for both training and testing phase can

	Method	CIFAR-10(C)					CIFAR-100(C)					Tiny-ImageNet(C)				
		Clean	Corr. Seve. All		Corr. Seve. 5		Clean	Corr. Seve. All		Corr. Seve. 5		Clean	Corr. Seve. All		Corr. Seve. 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
NI	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
NI	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
DA	Mixup*	**95.80**	80.40	-	-	-	**79.70**	54.20	-	-	-	-	-	-	-	-
DA	DeepAug*	94.10	85.33	64.63	77.29	60.05	-	-	-	-	-	**54.90**	-	-	-	-
DA	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
DA	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
AT	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
AT	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
AT	$RLAT_{AM}$	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
	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**

Table 1: Comparisons of PDE+ and baselines on CIFAR-10(C), CIFAR-100(C) and Tiny-ImageNet(C) based on ResNet-18. The corruption is evaluated under all severity level and the severest level. The best result is highlighted in boldface.The abbreviations means Standard (Std), Lipschitz (Lip), Noise Injection (NI), Data Augmentation (DA), Adversarial Training (AT).

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 2: Single source domain generalization comparisons of PDE+ and baselines on PACS datasets based on ResNet-18 (He et al. 2016). The best result is highlighted in boldface.

be found in Algorithms 1 and 2 in Appendix D.

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}} \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim s_N} -\log p_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}}(y \mid \mathbf{x}) = -\frac{1}{N}\sum_{n=1}^{N}\left[\log \frac{\exp(o_{\boldsymbol{\psi}}(\tilde{\mathbf{h}}_{n,L})_{y_n})}{\sum_{c=1}^{C}\exp(o_{\boldsymbol{\psi}}(\tilde{\mathbf{h}}_{n,L})_c)}\right]_{y_n} \tag{18}$$

5 Experiments

In this section, we empirically evaluates PDE+ through the following questions. Due to the space limitations, more comprehensive experiments including full results on corruptions and diffusion scale analysis are provided in Appendix E.

- (Q1) Does PDE+ improve generalization compared to SOTA methods on various benchmarks?
- (Q2) Does PDE+ learns appropriate diffusion distribution coverage?
- (Q3) Does PDE+ improve generalization beyond observed (training) distributions?

Experiments Settings A brief introduction of datasets, baselines and metrics is provided here, details can be found in Appendix F. **(1) Datasets**: Our experiments primarily focus on two types of datasets: (*i*) The original and 15 shift corruption distributions provided by CIFAR-10(C), CIFAR-100(C) and Tiny-ImageNet(C) (Krizhevsky, Hinton et al. 2009; Le and Yang 2015; Hendrycks and Dietterich 2019). (*ii*) The PACS dataset (Li et al. 2017) encompasses four different domains: photo, art, cartoon, and sketch. **(2) Baselines**: we consider the representative and SOTA methods as baselines: standard training; Lipschitz continuity based gradient regularization (Drucker and Le Cun 1992); Noise injection based methods, including EnResNet (2020), RSE (2018), NFM (2022); Data augmentation based methods, including Gaussian noise, Mixup (2018), DeepAug (2021), AutoAug (2019) and AugMix (2020); Adversarial training based methods, including PGD (2018) and RLAT (2022). **(3) Metrics**: Accuracy is adopted as the main evaluation metric. Especially, for various corrupted distributions, mCE (Hendrycks and Dietterich 2019) is adopted for two severity levels: severity across all levels and the severest level 5. More comprehensive results for other severity and metrics are in Appendix E. **(4) Others**: According to Section 4.2, the semantically similar samples in PDE+ are generated using AugMix (Hendrycks et al. 2020), a widely adopted data augmentation strategy that combines 7 distinct types of augmentations. It is important to note that we avoid overlap between these augmentations and the test distributions for most experiments.

5.1 Q1: PDE+ Outperforms SOTA on Benchmarks

Table 1 illustrates the results of PDE+ on CIFAR10(C), CIFAR100(C) and Tiny ImageNet(C) compared to baselines. "*" indicates that we reuse the results from Erichson et al. (2022) and Klim, Maksym, and Nicolas (2022). "-" indicates that this setting was not included in the paper. On original datasets, PDE+ achieves better performance than ERM, indicating that our diffusion does not obtain O.O.D. generalization at the cost of damaging performance on the original training distribu-

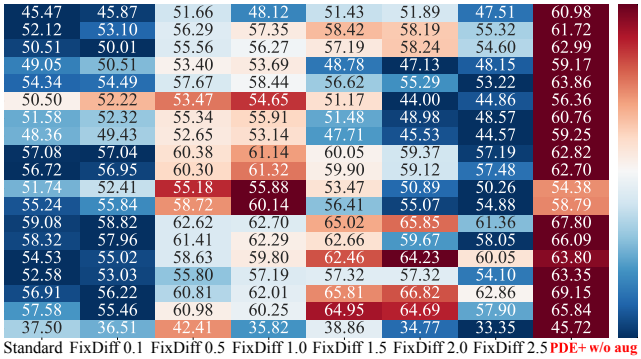

Figure 4: The heatmap of performance on neural networks with fixed diffusion (FixDiff) and PDE+ under fair comparison. The FixDiff scale is increasing from 0 to 2.5. The y-axis denotes 19 different test data distributions on CIFAR-10-C.

tion. The test distributions in corrupted datasets are different from training ones, which can be used to verify the effectiveness of generalization. Compared to numerous representative baselines across multiple categories, PDE+ achieves the best performance with respect to Acc and mCE on corruptions at the severest level and across all levels. The improvements are up to 3.8% in Acc and 7.7% in mCE. Such significant improvements make PDE+ stand out from other approaches that struggle to consistently improve performance across both original and diverse shifted distributions.

Table 2 illustrates the results of PDE+ on PACS datasets. When training on a single source domain and testing on the remaining 3 domains, PDE+ surpasses the baselines across all splits. This validates its efficacy not only in handling corruption data, where distribution shifts may be relatively close, but also demonstrates effectiveness with cross-domain data where distribution shifts can be significantly larger.

5.2 Q2: PDE+ Learns Appropriate Diffusion

This experiment is devoted to evaluating whether our proposed approach, whose diffusion scale is guided by augmented samples, can learn the appropriate diffusion scope. For a fair comparison, we do not conduct diffusion for augmented samples and only use augmented samples for the diffusion coverage guidance of the original samples (PDE+ w/o aug). This experiment can be viewed as the ablation study to evaluate if our learnable diffusion really works compared to fixed-scale diffusion (FixDiff for shorthand). Two conclusions can be drawn from the experimental results shown in Fig. 4: (1) Different corruption types, i.e., different distribution, prefers different magnitude/scale of smoothness, and a hard-to-please-everyone dilemma is caused by the fixed scale. (2) PDE+ indeed learns the appropriate diffusion scale. As is shown in the rightmost column, we can either achieve or be close to, the best performance of all corruption types.

5.3 Q3: PDE+ Generalizes Beyond Observation

This subsection aims to demonstrate that our method can generalize on distributions beyond training ones. Fig. 5 presents the changing trend of diffusion coverage for unobserved distributions, i.e., probabilities of unobserved test samples

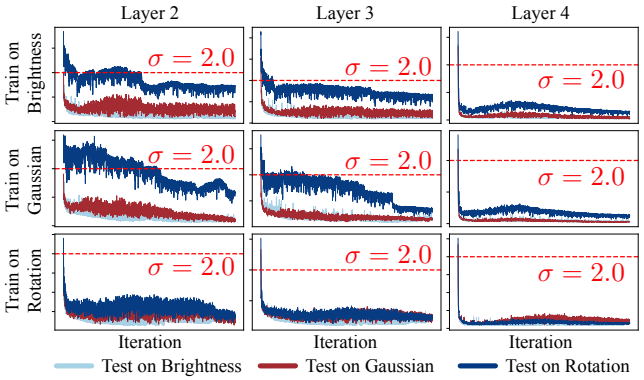

Figure 5: Diffusion coverage for unobserved distributions. Rows represent the training augmentations. Columns correspond to the layers of neural network. Each sub-figure includes three plots of distance-σ ratio during training for test samples generated by different test augmentations.

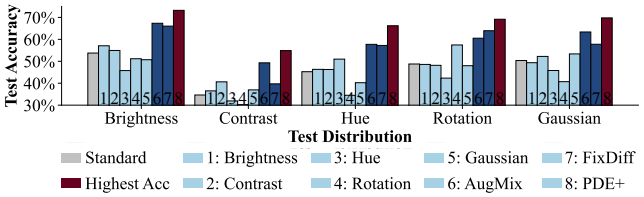

Figure 6: Generalization performance under five test distributions across eight different methods.

within the training diffusion distribution. This experiment is based on 2-σ rule of Gaussian distribution, detailed description can be found in Appendix E.4. The results imply that even when training occurs on a single augmentation differing from testing, the likelihood of test samples being perceived as normal within the training diffusion distribution increases over time. Fig. 6 represent the experiment as an extension of the previous one on Fig. 1. Notably, PDE+ outperforms all other augmentations, including AugMix, demonstrating its capability of generalization on unobserved distributions.

6 Conclusion

In conclusion, we present a novel partial differential equations (PDE)-driven approach to address the generalization issue of neural networks across unseen data distributions, focusing on overcoming the limitations of data-driven methods. By modeling neural networks as solutions to PDEs in a transport equation framework, the connection between the solution smoothness of PDEs and the generalization of neural networks is established. The introduction of an adaptive distributional diffusion term helps improve the generalization of neural networks. An instantiation of this framework, called PDE+ can enhance the generalization via taking the augmented samples as semantic similar samples to guide the learning of adaptive distributional diffusion. Experimental results demonstrate the superior performance of PDE+ across various shifted distributions. This work opens up new avenues for research in generalization of neural networks from the PDE perspective and offers a promising direction for enhancing the generalization of neural networks.

Acknowledgments

This work was supported by the National KeyR&D Program of China (2022YFB3103700,2022YFB3103704), the National Natural Science Foundation of China (NSFC) under Grants No.U21B2046 and No.62202448.

References

An, G. 1996. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3): 643–674.

Anil, C.; Lucas, J.; and Grosse, R. B. 2019. Sorting out Lipschitz function approximation.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Arrigo, D. J. 2017. *An Introduction to Partial Differential Equations*. Synthesis Lectures on Mathematics & Statistics. Morgan & Claypool Publishers.

Bishop, C. M. 1995. Training with Noise is Equivalent to Tikhonov Regularization. *Neural Comput.*, 7(1): 108–116.

Bousquet, O.; and Elisseeff, A. 2002. Stability and generalization. *The Journal of Machine Learning Research*, 2: 499–526.

Brezis, H.; and Brézis, H. 2011. *Functional analysis, Sobolev spaces and partial differential equations*, volume 2. Springer.

Butcher, J. C. 2003. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons.

Camuto, A.; Willetts, M.; Simsekli, U.; Roberts, S. J.; and Holmes, C. C. 2020. Explicit Regularisation in Gaussian Noise Injections. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 16603–16614. Curran Associates, Inc.

Chen, T.; Zhang, Z.; Liu, S.; Chang, S.; and Wang, Z. 2021. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*.

Cisse, M.; Bojanowski, P.; Grave, E.; Dauphin, Y.; and Usunier, N. 2017. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, 854–863. PMLR.

Cohen, S. B.; and Kirschner, I. N. 1991. Approximating the Dirac distribution for Fourier analysis. *Journal of computational physics*, 93(2): 312–324.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. AutoAugment: Learning Augmentation Strategies From Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Drucker, H.; and Le Cun, Y. 1992. Improving generalization performance using double backpropagation. *IEEE Transactions on Neural Networks*, 3(6): 991–997.

E, W. 2017. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 1(5): 1–11.

Eliasof, M.; Haber, E.; and Treister, E. 2021. PDE-GCN: novel architectures for graph neural networks motivated by partial differential equations. *Advances in neural information processing systems*, 34: 3836–3849.

Emmert-Streib, F.; and Dehmer, M. 2022. Taxonomy of machine learning paradigms: A data-centric perspective. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(5): e1470.

Erichson, N. B.; Lim, S. H.; Utrera, F.; Xu, W.; Cao, Z.; and Mahoney, M. W. 2022. Noisymix: Boosting robustness by combining data augmentations, stability training, and noise injections. *arXiv preprint arXiv:2202.01263*, 1.

Gelbrich, M.; and Römisch, W. 1995. Numerical Solution of Stochastic Differential Equations (Peter E. Kloeden and Eckhard Platen). *SIAM Rev.*, 37(2): 272–275.

Gold, S.; Rangarajan, A.; et al. 1996. Softmax to softassign: Neural network algorithms for combinatorial optimization. *Journal of Artificial Neural Networks*, 2(4): 381–399.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep learning*. MIT press.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations*.

Gulcehre, C.; Moczulski, M.; Denil, M.; and Bengio, Y. 2016. Noisy Activation Functions. In Balcan, M. F.; and Weinberger, K. Q., eds., *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, 3059–3068. New York, New York, USA: PMLR.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.

Haber, E.; and Ruthotto, L. 2017. Stable architectures for deep neural networks. *Inverse problems*, 34(1): 014004.

Hager, W. W. 1979. Lipschitz Continuity for Constrained Processes. *SIAM Journal on Control and Optimization*, 17(3): 321–338.

Han, J.; Jentzen, A.; and E, W. 2018. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34): 8505–8510.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; Song, D.; Steinhardt, J.; and Gilmer, J. 2021. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 8320–8329.

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*.

Hendrycks, D.; Mu, N.; Cubuk, E. D.; Zoph, B.; Gilmer, J.; and Lakshminarayanan, B. 2020. AugMix: A Simple Method to Improve Robustness and Uncertainty under Data Shift. In *International Conference on Learning Representations*.

Huster, T.; Chiang, C.-Y. J.; and Chadha, R. 2019. Limitations of the lipschitz constant as a defense against adversarial examples. In *ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings 18*, 16–29. Springer.

Jin, P.; Lu, L.; Tang, Y.; and Karniadakis, G. E. 2020. Quantifying the generalization error in deep learning in terms of data distribution and neural network smoothness. *Neural Networks*, 130: 85–99.

Kac, M. 1949. On Distributions of Certain Wiener Functionals. *Transactions of the American Mathematical Society*, 65(1): 1–13.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Klim, K.; Maksym, A.; and Nicolas, F. 2022. On the Effectiveness of Adversarial Training Against Common Corruptions. In *The 38th Conference on Uncertainty in Artificial Intelligence*.

Kloeden, P. E.; Platen, E.; Kloeden, P. E.; and Platen, E. 1992. *Stochastic differential equations*. Springer.

Kotz, S.; Kozubowski, T.; and Podgórski, K. 2001. *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. 183. Springer Science & Business Media.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. *Citeseer*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90.

Ladyzhenskaia, O. A.; Solonnikov, V. A.; and Ural'tseva, N. N. 1968. *Linear and quasi-linear equations of parabolic type*, volume 23. American Mathematical Soc.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, 5542–5550.

Li, M.; He, L.; and Lin, Z. 2020. Implicit Euler Skip Connections: Enhancing Adversarial Robustness via Numerical Stability. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Li, Z.; and Shi, Z. 2017. Deep Residual Learning and PDEs on Manifold. *CoRR*, abs/1708.05115.

Lim, S. H.; Erichson, N. B.; Utrera, F.; Xu, W.; and Mahoney, M. W. 2022. Noisy Feature Mixup. In *International Conference on Learning Representations*.

Liu, J.; Lin, Z.; Padhy, S.; Tran, D.; Bedrax Weiss, T.; and Lakshminarayanan, B. 2020. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33: 7498–7512.

Liu, W.; Wang, J.; Wang, H.; Li, R.; Qiu, Y.; Zhang, Y.; Han, J.; and Zou, Y. 2023. Decoupled Rationalization with Asymmetric Learning Rates: A Flexible Lipshitz Restraint. *arXiv preprint arXiv:2305.13599*.

Liu, X.; Cheng, M.; Zhang, H.; and Hsieh, C.-J. 2018. Towards Robust Neural Networks via Random Self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Lu, Y.; Zhong, A.; Li, Q.; and Dong, B. 2018. Beyond Finite Layer Neural Networks: Bridging Deep Architectures and Numerical Differential Equations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations*.

Munson, B. R.; Young, D. F.; Okiishi, T. H.; and Huebsch, W. W. 2006. Fundamentals of Fluid Mechanics, John Wiley & Sons. *Inc., USA*.

Ng, N.; Hulkund, N.; Cho, K.; and Ghassemi, M. 2022. Predicting Out-of-Domain Generalization with Local Manifold Smoothness. *CoRR*, abs/2207.02093.

Pogodaev, N. 2016. Optimal control of continuity equations. *Nonlinear Differential Equations and Applications NoDEA*, 23: 1–24.

Raghunathan, A.; Xie, S. M.; Yang, F.; Duchi, J.; and Liang, P. 2020. Understanding and mitigating the tradeoff between robustness and accuracy. *arXiv preprint arXiv:2002.10716*.

Reynolds, D. A.; et al. 2009. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663).

Rodríguez, P.; Laradji, I.; Drouin, A.; and Lacoste, A. 2020. Embedding propagation: Smoother manifold for few-shot classification. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, 121–138. Springer.

Rosca, M.; Weber, T.; Gretton, A.; and Mohamed, S. 2020. A case for new neural network smoothness constraints. In Zosa Forde, J.; Ruiz, F.; Pradier, M. F.; and Schein, A., eds., *Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops*, volume 137 of *Proceedings of Machine Learning Research*, 21–32. PMLR.

Sarra, S. A. 2003. The method of characteristics with applications to conservation laws. *Journal of Online mathematics and its Applications*, 3: 1–16.

Shorten, C.; and Khoshgoftaar, T. M. 2019. A survey on Image Data Augmentation for Deep Learning. *J. Big Data*, 6: 60.

Sokolic, J.; Giryes, R.; Sapiro, G.; and Rodrigues, M. R. D. 2017. Robust Large Margin Deep Neural Networks. *IEEE Trans. Signal Process.*, 65(16): 4265–4280.

Sun, Q.; Tao, Y.; and Du, Q. 2018. Stochastic Training of Residual Networks: a Differential Equation Viewpoint. *CoRR*, abs/1812.00174.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness May Be at Odds with Accuracy. In *International Conference on Learning Representations*.

Uhlenbeck, G. E.; and Ornstein, L. S. 1930. On the Theory of the Brownian Motion. *Phys. Rev.*, 36: 823–841.

Van Amersfoort, J.; Smith, L.; Teh, Y. W.; and Gal, Y. 2020. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*. PMLR.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold Mixup: Better Representations by Interpolating Hidden States. In *International Conference on Machine Learning (ICML)*, 6438–6447.

Wang, B.; Yuan, B.; Shi, Z.; and Osher, S. J. 2020. EnResNet: ResNets Ensemble via the Feynman–Kac Formalism for Adversarial Defense and Beyond. *SIAM Journal on Mathematics of Data Science*, 2(3): 559–582.

Wang, Y.-J.; and Lin, C.-T. 1998. Runge-Kutta neural network for identification of dynamical systems in high accuracy. *IEEE Transactions on Neural Networks*, 9(2): 294–307.

Xu, H.; and Mannor, S. 2010. Robustness and Generalization. In Kalai, A. T.; and Mohri, M., eds., *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, 503–515. Omnipress.

Yuan, Y.; Xu, B.; Lin, B.; Hou, L.; Sun, F.; Shen, H.; and Cheng, X. 2023. PDE+: Enhancing Generalization via PDE with Adaptive Distributional Diffusion. *arXiv preprint arXiv:2305.15835*.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; Ghaoui, L. E.; and Jordan, M. 2019. Theoretically Principled Trade-off between Robustness and Accuracy. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 7472–7482. PMLR.