Game Theory Meets Data Augmentation

Yuhan Kang, Samira Zare, Alex Lin, Zhu Han, Stanley Osher, and Hien Nguyen

***Abstract*—Data augmentation is a critical component in building modern deep-learning systems. In this paper, we propose `MFG Augment`, a novel data augmentation method based on the Mean-Field-Game (MFG) theory, that can synthesize a sequence of data between every two images or features. The central idea is to consider every image as a distribution over its pixel or feature space. Using Mean-field Game theory, we can generate a time-continuous "path" from one distribution to another so that the points along the "path" are augmented images or features. Empirically, the experiment results on MNIST, CIFAR-10, and ImageNet demonstrate that the proposed technology has better generalization ability and higher classification accuracy as compared to several benchmark methods. More importantly, our `MFG Augment` improves the test accuracy significantly when the dataset size is small. MFG Augment consistently shows better affinity and diversity scores, two important empirical metrics for evaluating the generalization of data augmentation techniques.**

***Impact Statement*—Data augmentation is a cornerstone in deep learning, essential for reducing overfitting by expanding training datasets. However, current data augmentation techniques face significant challenges. Many are tailored to specific data types, and lack a solid theoretical foundation linking generated and target data distributions. As a result, the model accuracy improvement is low. We propose a novel MFG-based data augmentation method: `MFG Augment`. The image's pixels or learned features are regarded as agents in our MFG model. Based on such, `MFG Augment` formulates the data augmentation as an optimization problem over the images' pixel space and feature space. From a theoretical standpoint, `MFG Augment` ensures a minimal divergence between the generated data's distribution and that of the original data. This is achieved using established mathematical constructs, namely the Hamilton-Jacobi-Bellman and Fokker-Planck-Kolmogorov equations, derived from MFG theory. We believe that MFG will pave the way for new applications by providing a data-agnostic, theoretically grounded, and highly effective data augmentation framework.**

***Index Terms*—Data Augmentation, Game Theory, Mean-field Game, Deep Learning, Image Classification**

I. Introduction

Data augmentation is proven to be a crucial technique for performing various deep learning tasks [1], [2], such as image classification [3], [4], [5], [6], speech recognition [7], natural language processing [8], etc. It aims to increase the amount of training data by adding slightly modified copies of existing data or newly created synthetic data from existing data. This mechanism helps learning models be more robust to common data variation. In the image domain, popular augmentation methods include translating the image by part of its pixels or flipping and rotating the whole image [9], [10], [3], [11], [12]. In the feature domain, common augmentations include interpolating learned features with linear transformation, adding noise, interpolating, and extrapolating between them [13], [14], or utilizing a deep neural network to model the feature trajectories [15], [16].

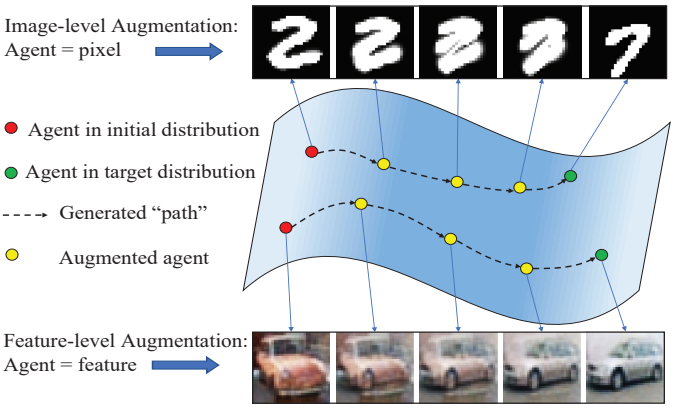


Fig. 1. Illustration of our `MFG Augment`: Idea 1: Image-level augmentation: Every image is regarded as a distribution over its pixel value space. The upper red point represents an image's pixel distribution. We use `MFG Augment` to transform from one image's distribution (red point) to another image's pixel distribution along a "path" (dotted line) so that the points along the path (yellow points) are augmented images. Idea 2: Feature-level augmentation: The whole dataset is regarded as a distribution over its feature space. We use `MFG Augment` to transform a distribution of a set of features (red point) to another distribution of a set of features (green point) along a path "dotted line", so that the points along the path (yellow points) are set of augmented features.

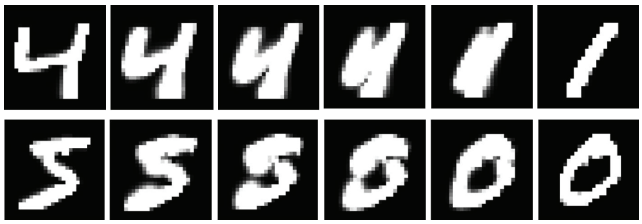
On the other hand, the Mean-field Game theory, an advanced variant of the optimal transport theory, has been proposed by Lasry and Lions in [17] and Caines and Huang in [18] in recent years. It is a mathematical framework developed to analyze situations where a large number of participants (or agents) make decisions based on their individual circumstances while considering the average behavior of the entire group. Here, the agents in the proposed `MFG Augment` refer to the pixels or the learned feature of an image. The basic idea of the mean-field game is that it can control the movement of the distribution of a large number of agents and transform it into a target distribution along an optimized "path". Different from traditional data augmentation methods that typically employ heuristic techniques, such as rotation, flipping, or cropping, without a unifying theoretical framework to guide and validate the quality and relevance of the generated data [19], `MFG Augment` offers a systematic and theoretical foundation to

This work is partially supported by NSF CNS-2107216, CNS-2128368, CMMI-2222810, ECCS-2302469, US Department of Transportation, Toyota. Amazon, JST ASPIRE JPMJAP2326, AFOSR FA9550-18-502, and ONR N00014-20-1-2787.

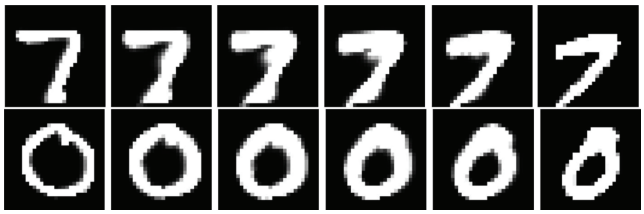
Y. Kang, S. Zare, and H. Nguyen are with the Electrical and Computer Engineering Department, University of Houston, Houston, TX, USA.

Z. Han is with the Department of Electrical and Computer Engineering at the University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul, South Korea, 446-701.

A. Lin and S. Osher are with the Department of Mathematics, University of California, Los Angeles, CA, 90095, USA.



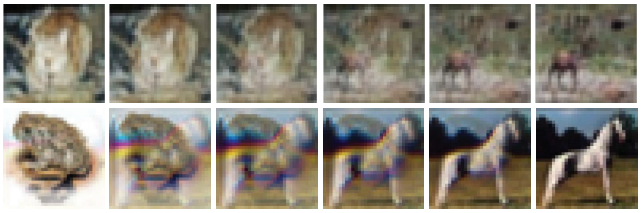
(a) Examples of label-variant `MFG Augment` in image-level augmentation. (Transform a "4" into "1", a "5" into "0").



(b) Examples of label-agnostic `MFG Augment` in image-level augmentation. (Transform a "7" into another "7", a "0" into another "0").

Fig. 2. Examples MFG Augment in image-level augmentation on MNIST.

(a) Examples of label-agnostic `MFG Augment` in feature-level augmentation. (Transform a bird into another bird, a car into another car.)



(b) Examples of label-variant `MFG Augment` in feature-level augmentation. (Transform a cat into a deer, a frog into a horse.)

Fig. 3. Visualization examples of MFG Augment in feature-level augmentation on CIFAR-10 (Decoded augmented features).

generate new data by modeling the relationship between original and augmented data distribution. Since its inception, MFG has achieved many successful applications in computer vision, manifold learning, reinforcement learning innovations, etc. For example, the famous Wasserstein Generative adversarial neural network (W-GAN) is based on the MFG theory to overcome the vanishing gradient problem [20]. It has also been used to find discriminant or robust subspaces for a dataset [21], [22]. Moreover, MFG theory can guide color transfer between images [23] or find correspondences between languages in word embeddings [24]. Other applications of MFG theory in engineering include robot swarm control [25], age of information minimization in wireless communication networks [26], resource allocation [27], data offloading optimization in edge computing [28], etc.

In this paper, we propose `MFG Augment`, a novel data augmentation method based on the MFG theory. We formulate the data augmentation process into a quantifiable MFG optimization problem in either the image's pixel space or feature space, and our approach can generate both augmented images and features. Examples of image-level `MFG Augment` and feature-level `MFG Augment` are shown in Fig. 2 and Fig. 3, respectively. To the best of our knowledge, this is the first work that integrates the Mean-field Game theory to data augmentation.

We demonstrate that the proposed `MFG Augment` has three main advantages: 1) The image-level MFG Augment is able to keep the shape and edge of the objects in images. 2) The feature-level augmentation is able to generate features in the learned manifold under quantifiable optimality conditions. 3) Empirically, our `MFG Augment` can improve the test accuracy significantly, especially for small datasets, improving from AugMix's 48.55training samples from CIFAR-10. Contributions of this work are summarized as follows.

- We formulate the data augmentation process as an MFG model based on the MFG theory framework, where the agent is modeled as either the pixel or the learned feature of an image. Based on the MFG model, we formulate the data augmentation process into an MFG optimization problem.
- To obtain the optimal augmentation strategy, we propose a G-prox PDHG algorithm and a GAN-like network to solve the proposed MFG data augmentation problem.
- We present a comprehensive analysis of the proposed `MFG Augment`, including optimality and time complexity analysis, all underpinned by the rigorous and theoretically-informed MFG theory.
- Experiments implemented on existing benchmark datasets demonstrate a significant improvement in test accuracy over state-of-the-art methods.

The empirical results on image classification tasks show that our method achieves significant improvements over existing data augmentation techniques, including `Cutout` [3], `CutMix` [11] and `Augmix` [12], on databases: MNIST, CIFAR-10, and ImageNet. Our source code will be made publicly available at https://github.com/YuhanK-A/MFG-Augment for research purposes. The rest of this paper is organized as follows. In Section II, we provide an extensive literature review in the field of data augmentation. Then we propose `MFG Augment` in Section III. In Section IV, we give a comprehensive analysis of the proposed `MFG Augment`. In Section V, we show the experiment results of the proposed `MFG Augment` by comparing them with several benchmarks. Finally, conclusions and future works are summarized in Section VI.

II. Related Work

Data augmentation stands as a cornerstone technique in enhancing the performance and generalization of machine learning models. Historically, data augmentation methods can be broadly categorized into two main classes based on the level at which they operate: image-level and feature-level. This categorization not only helps in understanding the evolution and

diversity of augmentation techniques but also contextualizes their application based on specific needs and challenges. We will discuss these two categories in Section II-A, and II-B, respectively.

A. Image-level Augmentations

Rotation, scale, translation, flip, elastic distortion, and jittering are common techniques used to augment image data [9], [10], [19], [29]. Random blocking techniques such as `Cutout` [3] that randomly masks out square regions of input images during training, `Hide-and-Seek` [30] that hides patches in a training image randomly, `Grid Mask` [31] that applies a grid-patterned mask comprised of regularly spaced rectangular regions onto input images, and `Random Erasing` [32] that involves erasing, or replacing, a randomly selected rectangular region within an input image, can also improve accuracy on clean data. Instead of randomly erasing images pixels, the authors in [6] use the saliency map to detect important regions on the original images and then preserve these informative regions during augmentation, which allows us to generate more faithful training examples. Rather than augmenting one image, some other techniques merge multiple input instances. For example, `Cutmix` [11] operates by cutting and pasting patches between two images and adjusting their corresponding labels proportionally. `Mixup` [33] combines two images using a convex combination. `SampleParing` [34] synthesizes a new sample from one image by overlaying another image. `BC Learning` [35] generates between-class images by mixing two images belonging to different classes with a random ratio. Also, `Co-Mixup` [36] maximizes the salient signal of input images and diversity among the augmented images. `CutPas` [37], together with some of its variants such as `Simple CutPas` [38] and `Continuous CutPas` [39], cuts object instances and paste them onto random backgrounds. Similarly, `FMix` [40] is also a mixed sample data augmentation technique that uses random binary masks obtained by applying a threshold to low-frequency images sampled from Fourier space. `SuperMix` [41] optimizes a mask for fusing two images to exploit the salient region with the Newton iterative method, which is 65x faster than gradient descent. `Scale and Blend` [42] cuts and scales object instances, and blends them in meaningful locations. `Context DA` [43] merges object instances using context guidance. `ClassMis` is proposed in [44] that generates augmentations by mixing unlabelled samples, by leveraging on the network's predictions for respecting object boundaries. Differently, the authors in [45] claim that non-label-preserving data augmentation can be surprisingly effective in improving model performance, and thus propose to use a non-linear combination of images to create augmented images.

Until now, the discussed augmentation techniques are model-free methods. However, various model-based data augmentation approaches, which utilize pre-trained models for generating augmented images, have likewise shown considerable effectiveness. For instance, in [46], [47], [48], a generative model is trained to do data augmentation, and in [49] the authors propose `BAGAN` as an augmentation tool to restore balance in imbalanced datasets. In [50], the authors train a CycleGAN to transform contrast CT images into non-contrast images, and then use the trained CycleGAN to augment training using these synthetic non-contrast images. `BDA` [51] uses CGAN to generate images using the Monte Carlo algorithm. `MFC-GAN` [52] handles the class imbalance problem by using multiple fake classes to obtain a fine-grained image for the minority classes. `ImbCGAN` [53] approximates the true data distribution and generate data for the minority classes of various imbalanced datasets. `AugGAN` [54] semantically preserves object when changing its style. Other types of data augmentation methods include using images with transferred style as in [55] that allows to generate the new images of high perceptual quality that combine the content of a base image with the appearance of another ones, and `StyleMix` [56] merges two images with style, content, and labels.

Besides those pixel modification methods, some methods learned augmentation policies, such as `AutoAugment` [4]. These methods search a group of augmentation operations to determine the optimal ones that optimize the downstream task's performance. A variant of `AutoAugment` named `Fast AutoAugment` is proposed to speed up the searching process using efficient density matching for augmentation policy search, `Faster AutoAugment` uses a differentiable policy search pipeline via approximate gradients, and `Adversarial AutoAugment` [57] that simultaneously optimizes target related object and augmentation policy search loss. Also, `Randaugment` is proposed in [5] that reduces the searching space that jointly controls all operations and hence reduces the computational expense of automated augmentation. `PBA` [58] adopts non-stationary augmentation policy schedules instead of a fixed augmentation policy via population-based training. `LSSP` [59] presents a policy-driven sequential image augmentation approach for image-related tasks. `AdaTransform` [60] uses a competitive task to obtain augmented images, in the training stage, a competitive task is employed to acquire augmented images with high task loss, and in the testing stage, a cooperative task is utilized to generate augmented images with low task loss. `CDST-DA` [61] utilizes a GAN to optimize a generative sequence, ensuring the transformed image aligns with the same class distribution. In [62], the authors use an influence function to predict how validation loss is affected by a particular training sample and minimize the approximated validation loss. `SPA` [63] automatically selects suitable samples for data augmentation. In addition, `Augmix` in [12] is characterized by the combination of multiple basic augmentation procedures paired with a consistency loss. These augmentation methods are chosen at random and stacked, resulting in a rich variety of enhanced images. However, these existing image-level augmentations may create ambiguous images or discard critical information from the original ones, making it difficult for a neural network to learn the shape features or localize important objects. Our `MFG Augment` can better maintain image structures during the augmentation process, as discussed in Section III-B3.

B. Feature-level Augmentations

In [14], the authors augment the learned feature space of the training dataset with simple transformations, such as adding noise, interpolating, or extrapolating between them. `Manifold Mixup` [13] extended `Mixup` by applying convex combination to the feature map. `Puzzle Mix` [64] and `FMix` [40] apply saliency maps and Fourier transform to use semantically representative parts of the data when generating features. On the other hand, some learning-based methods are also used to generate features. For example, `FATTEN` [15] utilizes a deep encoder/decoder network architecture to model the feature trajectories. In [16], a variational autoencoder-based feature augmentation method is proposed for acoustic modeling. In [65], `Moment Exchange` is proposed that replaces the moments of the learned features of one training image by those of another, and also interpolates the target labels. It forces the model to extract training signal from the moments in addition to the normalized features. `FeatMatch` [66] learns complex, feature-based transformations as well as incorporates information from class-specific prototypical representations. `SFA` [67] augments feature representation using random noise. In [68], the authors augment features using a generator by playing the GAN minimax game against source features. `TriNet` [69] is an auto-encoder network that is proposed to directly synthesize instance features. In [70], the authors propose a feature augmentation method based on the disentangled representation of intrinsic and bias attributes. `CRAFT` [71] is designed for e-identification tasks that performs cross-view adaptation by automatically measuring camera correlation from cross-view visual data distribution and adaptively conducting feature augmentation to transform the original features into a new adaptive space. Spectral feature augmentation is proposed in [72] for contrastive learning on graphs (and images) where the authors estimate a low-rank approximation for each data in the feature map and subtract that approximation from the map to obtain its complement. `A-FAN` [73] generates adversarial features that integrate flexible scales of perturbation strengths, and the features are re-injected through feature normalization. In [74], the authors propose a hyperbolic feature augmentation method that generates diverse and discriminative features in the hyperbolic space to combat overfitting.

On the other hand, semantic data augmentations are also used to complement traditional augmentation techniques. For example, in [75], [76], the authors propose to translate training samples along many semantic directions in the feature space. In [77], `Attribute Mix` is proposed to mix semantically meaningful attribute features from two images that can significantly improve the recognition performance without increasing the inference budgets. In [78], `ObjectAug` is proposed that decouples the image into individual objects and the background using semantic labels, and each object is augmented individually using traditional augmentation methods, and the augmented objects are finally assembled as augmented image. The authors in [79] propose a reasoning-based implicit semantic data augmentation method that samples new directions from similar categories based on estimated covariance matric for each category. `FASA` [80] generates virtual features on the fly to provide more positive samples for rare classes, and leverages a loss-guided adaptive sampling scheme to avoid over-fitting. Also, `AutoFeature` [81] augments the features following an exploration-exploitation strategy in a reinforcement learning framework. `MetAug` performs a meta-learning technique to augment the features by building the augmentation generator that updates its network parameters by considering the performance of the encoder.

III. MFG Augment

Image-level augmentations primarily cater to spatial transformations, providing the model with various visual perspectives and enhancing its robustness to different visual conditions. However, these might not always capture intricate data characteristics essential for some tasks. On the other hand, feature-level augmentations target the abstract representations learned by the model. By augmenting at this level, we ensure that the model is not just focusing on raw visual cues but also on the higher-level features that are crucial for accurate decision-making. By combining both levels of augmentation, we aim to create a comprehensive augmentation strategy that holistically improves model generalization, catering to both visual diversity and abstract feature enhancement. Therefore, in this section, we first give a brief introduction of the MFG theory in Section III-A, and then propose and discuss the details of the image-level `MFG augment` and feature-level `MFG augment` in Section III-B and Section III-C, respectively.

A. MFG Introduction

To begin with, we first give a brief introduction to the MFG theory. MFG is a mathematical framework that models the interactions of a large number of agents, where each agent's strategy depends on the statistical distribution of the strategies chosen by all other agents. The core idea of an MFG model is that by controlling the group strategy of all agents, the agents' state distribution evolves with time across an optimized path so that their costs that generate along the path can be minimized.

A toy example is that: Imagine a big crowd of people. While each person might decide to move in a certain way, MFG tries to understand how the entire crowd moves based on the average choices people make. This helps in predicting how large groups behave without getting lost in the details of each individual's actions. It's a way to understand big group dynamics without needing to know every tiny detail about everyone. In the proposed `MFG Augment`, the agent is an image's pixel or a learned feature. The agent state is the image's pixel value or the value of its learned feature, and the agent's strategy is a newly introduced variable that is used to change the image's pixel value or feature value.

Mathematically, a typical MFG model consists of a large number of agents denoted by $A = \{a_1, a_2, \ldots, a_k\}$, the agent dynamics is governed by the following stochastic equation:

$$dX^k(t) = f\left(u^k(t)\right)dt + \sigma dW^k(t),\ X^k(0) = x_0^k, \quad (1)$$

where $X^k(t)$ is the state of agent a_k, i.e., the "pixel" and "feature" at time t, $u^k(t)$ is the control input, and x_0^k is the

initial state of agent a_k at time $t = 0$. f is a function that describes how the agents' control inputs influence their states. $W^k(t)$ is a standard Brownian motion that captures the agent's stochastic property, and σ is its intensity. Under the dynamics constraint, agents aim to minimize the cost in the time interval $[0, T]$ by finding their optimal strategy $u^*(t)$:

$$\min_{u(t)} J_k = \mathbb{E}_X \left[\int_0^T L\Big(u(t), X^k(t), \rho(t)\Big) dt + G\Big(X^k(T), \rho(T)\Big) \right], \quad (2)$$

where $\rho(t)$ is the distribution of agents' state at time t, known as the mean-field term in MFG theory. Here, $L\left(u(t), X^k(t), \rho(t)\right)$ is known as the running cost, since it is generated continuously in the time interval $[0, T]$, and $G\left(X^k(T), \rho(T)\right)$ is known as terminal cost, since it is generated only at the terminal time T. The agents' optimal control strategy $u^*(t)$ is given by solving the mentioned MFG problem.

Remark 1: In the MFG theory, the mean-field term $\rho(t, x)$ serves as a variable that summarizes the state of all the agents. In other words, the relationship and interactions between pixels or the learned features are summarized in $\rho(t)$, which enables us to optimize the data augmentation process from a micro perspective, which is different from existing works that change each pixel or feature independently without considering the relationship between them.

B. Image-Level MFG Augment

Let $x_k(t) \in \mathbb{R}$, $k = 1, 2, \ldots, N$, be the k-th pixel of an image, where N is the image size. Then, an image can be represented as a distribution over its' pixel value space. **The idea is that:** we transform the distribution of an image's pixels, denoted as $\rho_{Initial}(x)$ into the distribution of another image's pixels $\rho_{Target}(x)$ within a time interval $[0, T]$ along a "path" $\rho^*(t, x)$, so that we can generate new images by sampling from the "path". Fig. 1 provides a visual illustration of the idea.

1) Problem Formulation of Image-Level MFG Augment: Since the initial point (i.e., when $t = 0$) is the initial image pixel distribution, we have:

$$\rho(0, x) = \rho_{Initial}(x). \quad (3)$$

On the other hand, to control the transformation direction of the "path", we define a control variable $u_k(t)$ that can change pixels' values, and the controlling process is described by the following equation:

$$dx_k(t) = u_k(t)dt. \quad (4)$$

To summarize the global information of all pixels, we define the mean-field term, i.e., the distribution of the image's pixels as $\rho(t, x)$. Then, according to [17], dynamics in (4) is transformed into its distribution dynamics:

$$\partial_t \rho(t, x) + \nabla \cdot (\rho(t, x) u(t, x)) = 0, \quad (5)$$

where ∂ is the partial derivative operator, and ∇ is the divergence operator. $u(t)$ is transformed into a field control $u(t, x)$.

We also avoid changing the image too fast along the path $\rho(t, x)$ so that the augmented images in adjacent time slots in the "path" share a certain similarity. Therefore, we quantify the cost function by imposing a penalty on the L2-norm of the control function. The overall MFG augmentation problem at the image level is given as follows:

$$\min_{u,\rho} J = \int_0^T \int_\Omega \rho(t, x) \|u(t, x)\|_2^2 \, dx dt + KL\left(\rho(T, x) \| \rho_{Target}(x)\right)$$
$$s.t. \quad \begin{cases} \mathcal{C}_1 : \partial_t \rho(t, x) + \nabla \cdot (\rho(t, x) u(t, x)) = 0, \\ \mathcal{C}_2 : \rho(0, x) = \rho_{Initial}. \end{cases} \quad (6)$$

where Ω is the pixel value space, i.e., $\Omega = [0, 255]$, of the images. KL is the Kullback–Leibler (KL) divergence that quantifies the distance between the final point in the path $\rho(T, x)$ and the target image's distribution ρ_{Target}, so that the path $\rho(t, x)$ can move towards the target distribution ρ_{Target}. The constraint $\mathcal{C}_1$ in Eq. (6) comes from Eq. (5) that describes how the augmented images ρ in the "path" are controlled by the control variable u, and the constraint $\mathcal{C}_2$ comes from Eq. (3) indicating that the start point of the augmented image path is the initial image's pixel distribution.

2) G-prox PDHG Algorithm: We utilize the numerical method: G-prox PDHG algorithm, proposed in [82] to solve the proposed image-level `MFG Augment` problem in (6). In summary, the algorithm transforms the MFG problem in Eq. (6) into a Lagrangian function, and then solves the transformed dual problem. The detailed steps are shown below.

Step 1: Define the Lagrangian function $\mathcal{L}$ based on Eq. (6) as:

$$\mathcal{L} = J - \int_0^T \int_\Omega \phi(t, x) \left(\partial_t \rho(t, x) + \nabla \cdot m(t, x)\right) dx dt, \quad (7)$$

where $m(t, x) = \rho(t, x) u(t, x)$ is an intermediate variable, $\phi(t, x)$ is the introduced dual variable, and J is the MFG cost

Algorithm 1 MFG Augment in Image-level Augmentation

1: **Input:** Randomly choose two images with paired labels (x_1, y_1), (x_2, y_2); Two possibility parameters $\alpha, \beta \in [0, 1]$; Two random generated variables $a, b \in [0, 1]$.
2: **if** $a \leq \alpha$ and $b \leq \beta$ **then**
3: **Initialize:** The initial image's pixel distribution $\rho_{Initial}$ by normalizing x_1, and target's image pixel distribution ρ_{Target} by normalizing x_2; Max iterations for G-prox PDHG algorithm K; Step size τ, σ; Terminal Time T.
4: **Define:** the Lagrangian function as in Eq. (7).
5: **while** $k \leqslant K$ **do**
6: update ρ, m, ϕ based on Eq. (9)
7: $k = k + 1$.
8: **end while**
9: **end if**
10: **Output:** The augmented images $\rho(t_i, x)$ are obtained by sampling the optimal image augmentation path $\rho^*(t, x)$ at time instances $t = t_1, t_2, \ldots, t_N$, where the label of the augmented image $\rho(t_i, x)$ is y_1 if $t_i \leq 0.5T$, and its' label is y_2 if $t_i \geq 0.5T$.

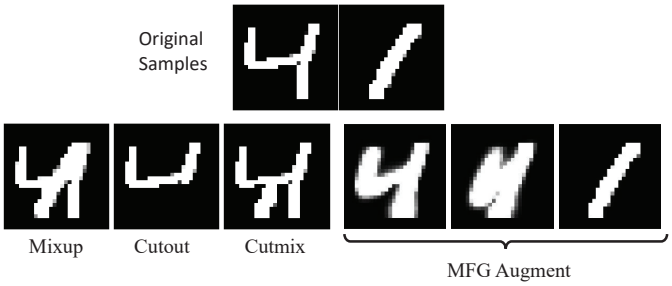


Fig. 4. Comparison of generated images with `Mixup`, `Cutout`, `Cutmix`, and `MFG Augment`. Note that `MFG Augment` can keep the edges and shape of objects in the generated images, but `Cutout`,`Cutmix`, and `Mixup` can not.

function defined in Eq. (6).

Therefore, the optimal image-level data augmentation control strategy $u^* = \frac{m^*}{\rho^*}$ is given by optimizing the three optimization variables:

$$\rho^*, m^*, \phi^* = \arg\min_{\rho,m}\max_{\phi}\{\mathcal{L}\} \tag{8}$$

Step 2: Update the augmented image path ρ, the intermediate variable m, and the introduced dual variable ϕ iteratively by solving the min-max problem in Eq. (8) based on the following iterative strategy.

$$\begin{cases} \rho^{k+1} = \arg\min_{\rho}\left\{\mathcal{L} + \frac{1}{2\tau}\left\|\rho - \rho^k\right\|_{L^2}^2\right\}, \\ m^{k+1} = \arg\min_{m}\left\{\mathcal{L} + \frac{1}{2\tau}\left\|m - m^k\right\|_{L^2}^2\right\}, \\ \phi^{k+1} = \arg\max_{\phi}\left\{\mathcal{L}(\bar{\rho}^{k+1}, \bar{m}^{k+1}, \phi^k) - \frac{1}{2\sigma}\left\|\phi - \phi^k\right\|_{H^1}^2\right\}, \end{cases} \tag{9}$$

where τ and σ are small step sizes. $\bar{\rho}^{k+1} = 2\rho^{k+1} - \rho^k$, $\bar{m}^{k+1} = 2m^{k+1} - m^k$, and L^2 and H^1 norm are defined as: $\|u\|_{L^2}^2 = \int_0^T\int_\Omega (u(t,x))^2\,dxdt$, $\|u\|_{H^1}^2 = \int_0^T\int_\Omega (\partial_t u(t,x))^2 + \|\nabla u(t,x)\|^2 dxdt$. The summary of the G-prox PDHG algorithm can be found in Algorithm 1.

3) What kind of images does image-level MFG Augment generate?: In MFG theory, the value of the control $u(t,x)$ is decided by the time t and pixel x. In other words, at the same time t, as long as two pixels have the same pixel value and the pixels with the same target pixel value, they will have the same control u. This property is beneficial for image augmentation since pixel with the same values usually belongs to one object, and pixels from adjacent locations with different values form an "edge". By using our `MFG Augment`, the generated image can keep the "edge shape" unchanged during the transformation. Fig. 4 compares `MFG Augment` with `Mixup`, `Cutout`, and `Cutmix`. One can notice that the generated image by `Mixup`, `Cutout`, and `Cutmix` is unnatural and can not keep the "edge" and "shape" as the original samples. However, our `MFG Augment` keeps a great shape with the original samples.

C. Feature-Level MFG Augment

We introduce `MFG Augment` in feature-level augmentation in this section. The idea is inspired by [83], [84], [14], where the authors claim that higher-level representations can expand the relative volume of plausible data points within the feature space. As such, when traversing along the learned manifold, it is more likely to encounter realistic samples in feature space than compared to the input space. Meanwhile, the development of self-supervised learning offers an effective way for learning useful features, e.g., variational autoencoders [85], generative adversarial networks [86], etc.

1) Problem Formulation of Feature-Level MFG Augment: Let $s_k(t) \in \mathbb{R}^m$, $k = 1, 2, \ldots, N$, be the learned features of the k-th image in a dataset, where N is the dataset size and t is time and m is the dimension of the feature. **The idea is that:** The whole dataset is regarded as a distribution over its low-dimensional feature space. We use `MFG Augment` to transform a distribution of a set of images' features that includes N_1 images, denoted as $\rho_{Initial}(s)$, to another distribution of a set of images' features that includes N_2 samples ($N_1 + N_2 = N$), denoted as $\rho_{Target}(s)$, along an optimized "path" $\rho^*(t,s)$ within a time interval $[0,T]$, so that the sampled points in such "path" are all augmented features. Ideally, the optimized "path" is the exact manifold of the images in their latent space. Fig. 1 describes a conceptive illustration of the idea.

Since the initial point ($t = 0$) and the final point ($t = T$) of the path $\rho(t,s)$ is the distribution of the initial set of features and the distribution of the target set of features, we have:

$$\rho(0,s) = \rho_{Initial}(s), \tag{10}$$
$$\rho(T,s) = \rho_{Target}(s). \tag{11}$$

On the other hand, to control the transformation direction of the "path", we also define a control variable $u(t)$ that can control the flow of the distribution of features. The controlling process is described by,

$$ds_k(t) = u_k(t)dt. \tag{12}$$

Similar to Eq. (5), in order to summarize the global information of all learned features in the feature space, we again define the mean-field term, i,e,m the distribution of the learned features as $\rho(t,s)$. According to [17], the state-control dynamics of the augmented feature distribution is given by:

$$\partial_t\rho(t,s) + \nabla\cdot(\rho(t,s)u(t,s)) = 0. \tag{13}$$

Similar to image-level augmentation, we aim to generate a "smooth" path between two distributions. We also impose an L-2 norm penalty on the control in the cost function. Therefore, the overall feature-level `MFG Augment` problem is given as:

$$\min_{u,\rho} J = \int_0^T\int_\Omega \rho(t,s)\left\|u(t,s)\right\|_2^2\,dsdt \tag{14}$$

$$s.t. \quad \begin{cases} \mathcal{C}_1 : \partial_t\rho(t,s) + \nabla\cdot(\rho(t,s)u(t,s)) = 0, \\ \mathcal{C}_2 : \rho(0,s) = \rho_{Initial}, \\ \mathcal{C}_3 : \rho(T,s) = \rho_{Target}, \end{cases} \tag{15}$$

where Ω is the image's feature space (the latent space). The constraint $\mathcal{C}_1$ comes from Eq. (13) which describes how the augmented feature distribution ρ is controlled by the introduced control variable u. The constraint $\mathcal{C}_2$ and $\mathcal{C}_3$ come from Eq. (10) and (11) indicating that the start point and the end point of the augmented feature path is the distribution of the initial set of features and the distribution of the target set of features.

2) APAC-Net Network to Solve Feature-Level MFG Augmentation Problem: The MFG problem in Eq. (14) is a high-dimensional problem because the learned features are usually hundreds or thousands-dimensional. According to [87], the grid-based numerical methods, such as PDHG and Adjoint method [88], are prone to the curse of dimensionality, i.e., their computational complexity grows exponentially with the spatial dimension. Thus, we utilize the APAC-Net, an alternating population and agent control neural network approach geared toward high-dimensional MFG problems [89] to solve it. In particular, we solve Eq. (14) by training a Wasserstein Generative Adversarial Network (WGAN)-like network. A detailed description of the implementation of the APAC-Net is shown below.

First, we transform the MFG problem into the Lagrangian problem:

$$\min_{\rho,u}\max_{\phi}\{\mathcal{L}\} = \int_0^T\int_\Omega \rho(t,s)\left\|u(t,s)\right\|_2^2 - \phi(t,x)\left(\partial_t\rho(t,s) + \nabla\cdot(\rho(t,s)u(t,s))\right)dsdt. \tag{16}$$

Second, we solve (16) by training a WGAN-like neural network named APAC-Net.

The neural networks generator is denoted by $G_\theta(s,t)$, and the discriminator is denoted by $N_\omega(s,t)$. The generator generates the path (i.e., the manifold), and the discriminator judges whether the generated path satisfies the optimality condition. Then we set

$$\phi_\omega(z,t) = (1-t)N_\omega(z,t), \tag{17}$$

$$G_\theta(s,t) = (1-t)s_0 + t(1-t)N_\theta(s,t) + ts_1, \tag{18}$$

where $s_0 \sim \rho_{Initial}$ are samples drawn from the initial distribution $\rho_{Initial}$, and $s_1 \sim \rho_{Target}$ are samples drawn from the target distribution. Note that the formulation of $G_\theta(s,t)$ automatically encodes the boundary conditions in Eq. (10) and (11).

During the training process of the APAC-Net, we alternatively train $G_\theta(s,t)$ and $\phi_\omega(s,t)$. Specifically, we first sample a batch $\{s_0^b\}_{b=1}^B$ from the initial distribution, a batch $\{s_1^b\}_{b=1}^B$ from the target distribution ρ_{Target}, and $\{t^b\}_{b=1}^B$ uniformly from the time interval $[0,T]$. Then we compute the output of the generator $z_b = G_\theta(s,t)$ as in Eq. (18). The total loss of the discriminator ϕ_ω is then given as the value function:

$$l_\phi = \frac{1}{B}\sum_{b=1}^B \phi_\omega(z_b,0) + \partial_t\phi_\omega(z_b,t_b) - |\nabla_z\phi_\omega(z_b,t_b)|^2, \tag{19}$$

where we can optionally add a regularization term that gives the optimality Karush–Kuhn–Tucker (KKT) condition of Eq. (14), which is also known as the Hamilton-Jacobi-Bellman (HJB) equation in MFG theory:

$$l_{HJB} = \lambda\frac{1}{B}\sum_{b=1}^B \left\|\partial_t\phi_\omega(z_b,t_b) - |\nabla_z\phi_\omega(z_b,t_b)|^2\right\|, \tag{20}$$

to penalize the derivations from the HJB equation. Finally, we back-propagate the total loss to update the weights of the discriminator ϕ_ω. To train the generator, we again sample $\{s_0^b\}_{b=1}^B$, $\{s_1^b\}_{b=1}^B$, and $\{t^b\}_{b=1}^B$ as before, and compute the loss of the generator:

$$l_G = \frac{1}{B}\sum_{b=1}^B \partial_t\phi_\omega\left(G_\theta(s_b,t_b),t_b\right) - |\nabla_z\phi_\omega(z_b,t_b)|^2. \tag{21}$$

At last, we back-propagate the loss to the weights of G_θ.

Once the optimal path $\rho^*(t,s)$ (i.e, the manifold) is generated, augmented features can be obtained by sampling $\rho^*(t,s)$ at any time $t \in [0,T]$. The augmented features can be used as input for a learning task or as the basis for generating new images. The training process of the APAC-Net is described in Algorithm 2.

Remark 2: In the image-level MFG augmentation problem in Eq. (6), we use the G-prox PDHG algorithm to solve it because as a numerical method, it can guarantee the convergence and optimality of the solution. However, the G-prox PDHG algorithm does not scale well to high-dimensional MFG problems as the feature-level optimization problem in Eq. (14), so here we train the APAC-Net network to solve it. On the other hand, although APAC-Net is able to solve high-dimensional MFG problems, it can not guarantee the convergence and optimality of its solution. Therefore, for the low-dimensional image-level MFG augmentation problem in Eq. (6), we use the G-prox PDHG algorithm to solve it.

IV. Theoretical and Performance Analysis of MFG Augment

In Section IV-A, we first analyze why `MFG Augment` generates "High-Quality" data by introducing the physical meaning of the proposed data augmentation optimization problem in Eq. (6) and (14). Then we propose a quantifiable metric for the proposed data augmentation process in Section IV-B, and give the complexity analysis in Section IV-C, respectively.

A. Analysis of the Proposed MFG Data Augmentation Optimization Problem

In Eq. (6) and (14), the physical meaning of the term $\int_0^T\int_\Omega \rho(t,x)\left\|u(t,x)\right\|_2^2 dxdt$ is the Wasserstein-2 (W-2) distance between the generated data distribution and the source data distribution, and we aim to minimize such W-2 distance. In existing works, the idea of minimizing the distance between generated data distribution and source distribution has been widely utilized in the design of Generative Adversarial Network (GAN) [90]. However, there exists the issue of Perfect Discriminator and Unstable Training since the loss function of GAN is a variation of Kullback–Leibler (KL) divergence, which is meaningless when two distributions have

Algorithm 2 MFG Augment in feature-level Augmentation

Input: The neural networks G_θ and ϕ_ω; Two sets of features with paired labels $(s_i^n, y_i^n)_{i=0,1}$; Two possibility parameters $\alpha, \beta \in [0,1]$; Two random generated variables $a, b \in [0,1]$.
if $a \leq \alpha$ and $b > \beta$ **then**
Initialize: G_θ and ϕ_ω as in (17) and (18); The initial distribution of features $\rho_{Initial}$ by scaling s_0^n; The target distribution of features ρ_{Target} by scaling s_1^n.
while not converge **do**
Train ϕ_ω for N epochs:
Sample batch (s_0^b, t_b), (s_1^b, t_b) where $s_0^b \sim \rho_{Initial}$, $s_1^b \sim \rho_{Target}$, and $t^b \sim \text{Unif}(0, T)$.
$z_b \leftarrow G_\theta(s_b, t_b)$ for $b = 1, 2, \ldots, B$
$l_\phi \leftarrow \frac{1}{B}\sum_{b=1}^{B} \phi_\omega(z_b, 0) + \partial_t\phi_\omega(z_b, t_b) - |\nabla_z\phi_\omega(z_b, t_b)|^2$
$l_{HJB} \leftarrow \lambda \frac{1}{B}\sum_{b=1}^{B} \left\| \partial_t\phi_\omega(z_b, t_b) - |\nabla_z\phi_\omega(z_b, t_b)|^2 \right\|$
Back-propagate the loss $l_{total} = l_\phi + l_{HJB}$ to weights ω.
Train G_θ M epochs:
Sample batch (s_0^b, t_b), (s_1^b, t_b) where $s_0^b \sim \rho_{Initial}$, $s_1^b \sim \rho_{Target}$, and $t^b \sim \text{Unif}(0, T)$
$l_G \leftarrow \frac{1}{B}\sum_{b=1}^{B} \partial_t\phi_\omega\left(G_\theta(s_b, t_b), t_b\right) - |\nabla_z\phi_\omega(z_b, t_b)|^2$
Back-propagate the loss l_G to weights θ.
end while
end if
Output: The augmented features set $\rho(t_i, s)$ are obtained by sampling the optimal path $\rho^*(t, s)$ at time instances $t = t_1, t_2, \ldots, t_N$, where the label of the augmented features set $\rho(t_i, s)$ is y_1 if $t_i \leq 0.5T$, and its' label is y_2 if $t_i \geq 0.5T$.

no overlap (i.e., log(0) is meaningless), and thus results in gradient vanishing [91]. To deal with the hard training of GAN, Wasserstein GAN [20] is proposed that utilize the Wassterin distance in the loss function instead of KL divergence. Compared with KL divergence, Wasserstein distance is symmetric and is able to quantify the distance between two distributions even if they have no overlap [92]. Therefore, in the proposed `MFG Augment`, the utilization of W-2 distance in the MFG framework enables us to generate data that has the smallest W-2 distance with source data (if the solution that we obtain is optimal). In addition, the utilization of MFG theory allows us to quantify the "quality" of the generated data distribution using the pre-defined Hamilton-Jacobi-Bellman (HJB) and the Fokker-Planck-Kolmogorov (FPK) equation, which is introduced in Section IV-B.

B. Quantifiable Optimality Condition of the Proposed MFG Augment Scheme

In the proposed `MFG Augment`, we formulate the data augmentation process as an MFG problem, and such formulation enables us to evaluate the performance of such data augmentation process. Specifically, in order to calculate the optimal data augmentation strategy $u(t, s)$ and $u(t, x)$ in Eq. (6) and (14), the following partial differential equation (PDE) should be satisfied:

$$\partial_t\rho - \frac{\sigma^2}{2}\Delta\rho - \nabla \cdot (\rho \nabla_p H(x, \rho, \nabla\phi)) = 0, \tag{22}$$

$$-\phi_t - \frac{\sigma^2}{2}\Delta\phi + H(x, \rho, \nabla\phi) = 0, \tag{23}$$

$$\rho(0, x) = \rho_0(x), \ \phi(T, x) = G\left(X(T), \rho(T, x)\right), \tag{24}$$

where Eq. (22) is named the Fokker-Planck-Kolmogorov (FPK) equation as introduced in Eq. (5). Eq. (23) is called the Hamilton-Jacobi-Bellman (HJB) equation that gives the optimality condition of the data augmentation process. Eq. (24) is the boundary condition of the PDE system.

Traditional metrics often evaluate data augmentation techniques based on heuristic measures such as test accuracy, which might not fully capture the specific optimality conditions that the proposed MFG Augment. However, given the theoretical foundation of our method in MFG, it is essential to have a metric that is compatible with its underlying principles. Here, we utilize the FPK and the HJB equation as a metric to evaluate the performance of the proposed data augmentation process, and the introduction of such quantifiable metrics is driven by the distinct nature of our proposed data augmentation method. Empirically, we give the residual of the FPK and the HJB equation using image-level `MFG Augment` on Mnist in Fig. 5a using Algorithm 1. It can be observed that after about 1,000 iterations the residuals become lower than 0.0001. Similarly, in 5b, the residual of the HJB equation is depicted using feature-level `MFG Augment` on CIFAR-10. It is shown that after about 10,000 training epochs the HJB residual becomes lower than 0.0001, which serves as a quantifiable metric to evaluate the performance of the data augmentation process besides model accuracy.

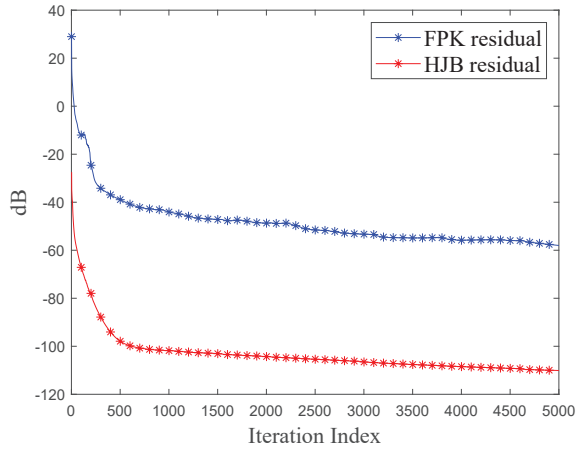
C. Complexity Analysis

In Algorithm 1, to find the optimal data augmentation strategy $u(t, x)$, the variables: $\rho(t, x)$, $m(t, x)$, and $\phi(t, x)$ are updated for each time step and pixels. Therefore, the computational complexity of the image-level `MFG Augment` is $\mathcal{O}(N \times D)$, where N is the number of time discretization slots, and D is the image size. Empirically, the time utilized to obtain optimal data augmentation strategies for image-level `MFG Augment` is $20 \sim 40$ seconds between every two images for Mnist, $40 \sim 60$ seconds for CIFAR-10, and $8 \sim 9$ mins for ImageNet. On the other hand, in Algorithm 2, we train a GAN-like network to find the feature-level augmentation strategy. Empirically, the time utilized to obtain optimal data augmentation strategies for feature-level `MFG Augment` is: $1.4 \sim 1.6$ hours for MNIST, $1.8 \sim 2.0$ hours for CIFAR-10, and $3.0 \sim 3.4$ hours for ImageNet, respectively. The experiments are implemented using CPU i5-12600 and GPU NVIDIA K80.

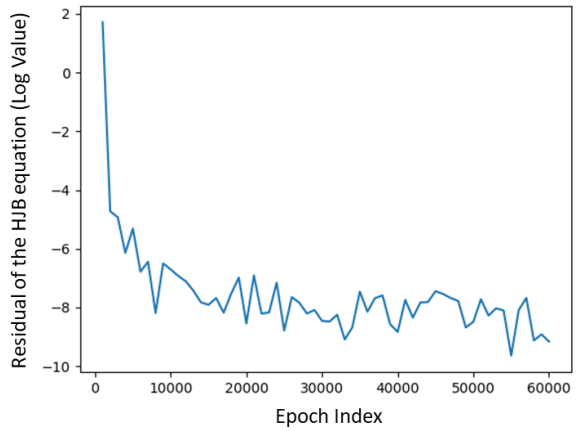
It is worthwhile to mention that image-level `MFG Augment` is designed to generate augmented images between every two original images, while feature-level `MFG Augment` is used to generate augmented features between every two random sets of learned features. In other words, Algorithm 1 needs to be repeated between every two original images,

Reduced trainset size	Baseline	Cutout	Cutmix	Augmix	MFG Augment
50	11.35	59.41	57.18	67.01	**70.61**
100	31.23	65.95	68.53	77.68	**80.26**
200	64.83	73.67	78.35	86.8	**88.94**
400	75.33	87.13	84.59	92.34	**94.95**
600	82.80	90.92	90.61	93.9	**96.34**
800	91.40	93.16	92.02	95.95	**97.96**
2,000	94.32	96.03	96.18	97.35	**98.52**

TABLE I
TEST ACCURACY (%) ON REDUCED MNIST WITH TRAINSET SIZES VARYING FROM 50 - 2,000, TRAINED WITH EFFICIENTNET. NOTE THAT THE FULL MNIST HAS 60,000 SAMPLES.



(a) Residual of the FPK and the HJB equation in image-level Augmentation, implemented on Mnist.



(b) Residual of the HJB equation in image-level Augmentation, implemented on CIFAR-10.

Fig. 5. Residuals of the FPK and HJB equation in `MFG Augment`.

Model	EfficientNet	ResNet-18	ResNet-50
Cutout	96.03	97.66	96.98
Cutmix	96.18	96.77	96.40
Augmix	97.35	98.50	98.16
MFG Augment	**98.52**	**99.02**	**98.62**

TABLE II
TEST ACCURACY (%) ON REDUCED MNIST, TRAINSET SIZE = 2000. NOTE THAT THE ORIGINAL MNIST HAS 60,000 SAMPLES.

while the GAN in Algorithm 2 is trained once on the whole original dataset distribution, which makes the image-level `MFG Augment` is more suitable for small datasets and feature-level `MFG Augment` are more efficient for large datasets.

V. EXPERIMENTS

In this section, we empirically evaluate our data augmentation method `MFG Augment`, and compare it with existing state-of-art data augmentation methods including `Cutout` [3], `Cutmix` [11], and `Augmix` [12]. These methods are widely recognized and established in the domain, offering diverse augmentation paradigms from occlusion-based to region-mixing and complex blending of augmentations. By comparing against these recognized techniques, our evaluation gains rigor and relevance. We implement the augmentation methods on datasets: MNIST [93], CIFAR-10 [94], and ImageNet [95] on several neural network architectures including EfficientNet [96], ResNet-18 [97], and Resnet-50 [97]. All the experiments are carried 5 times, and the average performance is reported. In summary, we observe significant improvements in our `MFG Augment` over the three comparative augmentation methods on both MNIST and CIFAR-10 for all tested neural network architectures.

A. MFG Augment Implementation Settings:

For the implementation of our `MFG Augment`, we combine both the image-level augmentation and the feature-level augmentation. We combine the augmented images, decoded augmented features, and the original dataset into the new trainset. Specifically, the two possibility parameters in Algorithm 1 and Algorithm 2 are set as $\alpha = 0.4$ and $\beta = 0.2$, respectively. That is, given two random data, the possibility that we implement our `MFG Augment` is 0.4, and the possibility that we implement image-level augmentation and feature-level augmentation is 0.2 and 0.8, respectively. The probability for data augmentation $\alpha = 0.4$, is determined through preliminary tests that balance data diversity with the original data's integrity. Also, we allocate different probabilities to image-level versus feature-level augmentations (i.e., $\beta = 0.2$), which is designed to maximize our method's effectiveness. In addition, we set the time interval $[0, T]$ to be $[0s, 1s]$ for both image-level and feature-level implementation, and we generate the augmented images and features by sampling 2 time-instances for every generated path $\rho^*(t, x)$ at time instants $t = 0.23s, 0.77s$. (Theoretically, we can sample an infinite number of augmented data since $\rho^*(t, x)$ is time-continuous. Here, the choice of the

Reduced trainset size	Baseline	Cutout	Cutmix	Augmix	MFG Augment
100	17.26	18.10	20.25	21.20	**29.21**
400	26.78	30.52	30.55	32.93	**52.51**
800	31.33	31.43	33.97	37.50	**61.14**
2,000	41.16	41.01	43.60	48.55	**65.11**
10,000	69.89	71.44	69.49	74.99	**83.23**
20,000	79.5	79.23	79.76	82.77	**86.87**
30,000	83.33	83.73	84.42	85.95	**88.50**
40,000	85.09	85.15	87.17	87.93	**89.95**
50,000	86.46	88.13	88.94	88.95	**90.43**

TABLE III
TEST ACCURACY (%) ON CIFAR-10 USING SEVERAL REDUCED TRAINSET WITH DATASIZE VARY FORM 100 - 50,000, TRAINED WITH EFFICIENTNET.

Model	EfficientNet	ResNet-18	ResNet-50
Cutout	88.13	88.27	89.61
Cutmix	88.94	91.37	92.49
Augmix	88.95	91.34	91.95
MFG Augment	**90.43**	**94.76**	**94.69**

TABLE IV
TEST ACCURACY (%) ON CIFAR-10.

two time-instants at $t = 0.23s, 0.77s$ is determined by a series of preliminary test, and such settings achieves the optimal performance for our proposed `MFG Augment` method). For the step sizes in Algorithm 1, we set $\tau = 9.9,\ \sigma = 0.1$, and the max iteration is set to be $K = 3,000$. For the training of the APAC-Net in Algorithm 2, we use a ADAM optimizer with $\beta = (0.5, 0.9)$, learning rate 0.0005 for ϕ_ω, learning rate 0.0001 for G_θ, weight decay of 0.0001 for both ϕ_ω and G_θ.

B. MNIST

1) Cutout, Cutmix, Augmix Implementation Settings: For a fair and clear comparison, we do not use other data augmentation settings such as random-cropping, or flipping. For `Cutout`, the paste-back size is set to be 8. For `Cutmix`, the hyperparameter: cutmix probability, is set to be 0.5, and the beta distribution $\beta(\alpha, \alpha)$, is set to be $\alpha = 1.0$, as suggested in [11]. For Augmix, the augmentation severity is set to be 2. Other augmentation hyperparameters follow the settings in [3], [11], [12].

2) Training settings: We first implement augmentations on the state-of-art neural network architecture: EfficientNet [96]. We optimize it with stochastic gradient descent using Nesterov momentum and train it for 100 epochs. The initial learning rate is set to be 0.01 [98]. Following [11], [12], we use a weight decay of 0.0001 for `Cutmix` and `Augmix`, and 0.0005 otherwise. Other hyperparameters follow the settings in [96].

We first train the EfficientNet with several reduced MNIST datasets with a variety of trainset samples from 50 to 2,000, and compare the test accuracy of the proposed `MFG Augment` with `Cutout`, `Cutmix`, and `Augmix` on these reduced MNIST trainsets.

3) Results: As is shown in Table I, we observe that our `MFG Augment` consistently and significantly improve over `Cutout`, `Cutmix`, and `Augmix` with all reduced trainsets. Specifically, when the data size of the reduced MNIST trainset is 50, our MFG Augment achieves the test accuracy gain $+59.26\%$ compared with the baseline and even achieves the accuracy gain at +11.2%, +13.2%, and +3.6% over `Cutout`, `Cutmix`, and `Augmix`, respectively. In addition, we can achieve a test accuracy of 98.52% even by training a reduced MNIST trainset with only 2,000 samples, while the standard MNIST needs to train all the 60,000 samples to get a comparable test accuracy.

We also compare the test accuracy on other state-of-art neural network architectures: ResNet-18 [97], and ResNet-50 [97]. All the hyperparameter settings are the same as in EfficientNet, except we train both networks for 150 epochs to guarantee convergence. The trainset is still the reduced MNIST with a data size of 2,000. As can be observed in Table II, our `MFG Augment` achieves the highest test accuracy over all three comparative augmentation methods. Specifically, we achieve the test accuracy 99.02% on ResNet-18, which is +1.36% better than `Cutout`, +2.25% better than `Cutmix`, and +0.52% better than `Augmix`. Also, on EfficientNet and ResNet-50, we achieve state-of-art test accuracy of 98.52% and 98.62% on reduced MNIST (trainset size=2,000), respectively.

C. CIFAR-10

1) Cutout, Cutmix, and Augmix Implementation Settings: For a fair and clear comparison, we do not use other data augmentation settings such as random cropping or flipping. For `Cutout`, we set the cutout paste-back size to be 16, as suggested by [3]. For `Cutmix`, the hyperparameter: cutmix probability, is set to be 0.5, and the beta distribution $\beta(\alpha, \alpha)$, is set to be 1.0, as suggested in [11]. For `Augmix`, the augmentation severity is set to be 3. All the other augmentation hyperparameters follow the settings in [3], [11], [12].

2) Training settings: Similar to the above experiments, we first train the EfficientNet with several reduced CIFAR-10 trainsets with trainset sizes varying from 100 to 50,000. The training settings, such as momentum, scheduler, etc., are the same as in the MNIST experiment, except the initial learning rate is 0.1, and we train the neural network for 250 epochs for convergence.

3) Results: As shown in Table III, we again observe consistent improvements in our `MFG Augment` over the three comparative augmentation methods. Especially, our `MFG Augment` gains impressive results when the dataset size is small.

Reduced trainset size	Baseline	Cutout	Cutmix	Augmix	MFG Augment
1/4 ImageNet	54.18	55.07	57.57	57.03	**61.13**
1/2 ImageNet	60.92	61.39	61.97	62.77	**65.22**
3/4 ImageNet	65.75	67.04	69.65	68.91	**69.73**
Full ImageNet	69.76	70.25	71.52	71.49	**72.26**

TABLE V
TOP-1 ACCURACY (%) ON SEVERAL REDUCED IMAGENET, TRAINED WITH RESNET-18.

When the trainset size is 2,000, our MFG Augment achieves 65.11% test accuracy, beating the Cutout 41.01%, Cutmix 43.60% and Augmix 48.55%. We again observe significant improvements when trained with all the 50,000 training samples. Specifically, our MFG Augment can achieve an accuracy of 90.43%, which is +3.97% higher than the baseline, and +2.30% higher than Cutout, +1.49% higher than Cutmix, and +1.48% higher than Augmix.

Again, we implement the augmentation methods on ResNet-18 and ResNet-50. We train with all the 50,000 training samples in CIFAR-10. The hyperparameter settings are the same as the previous EfficientNet of CIFAR-10, except we train the networks for 300 epochs for convergence. The results are shown in Table IV. Our MFG Augment significantly outperforms three comparative augmentation methods on all the tested neural network architectures. Specifically, on ResNet-18, MFG Augment achieves 94.76% test accuracy, which is +6.49% higher than Cutout, +3.39% higher than Cutmix, and +3.42% higher than Augmix. On EfficientNet and ResNet-50, our MFG Augment again achieves the state-of-art test accuracy at 90.43% and 94.69%, respectively.

D. ImageNet

1) Cutout, Cutmix, and Augmix Implementation Settings: All images are first pre-processed with standard random cropping horizontal mirroring. For Cutout, the mask size is set to 112×112, and the location for dropping out is uniformly sampled. For Cutmix, the cutmix probability is set as 1.0, and the beta distribution $\beta(\alpha, \alpha)$ is set to be 1.0, as suggested in [11]. For Augmix, the augmentation severity is set to 1.0, and the depth and width of augmentation chains are set to -1 and 3, as suggested in [12].

2) Training settings: We first train several reduced ImageNet trainsets with ResNet-18, and we follow the standard scheme of [97]. Then, we train the full ImageNet with Resnet-50 scheme of [97]. We found that Cutout and Cutmix require many training epochs to converge. Therefore, we optimize all the models using 300 epochs with an initial learning rate 0.1, decaying by 0.1 every 75 epochs. The batch size is 256.

3) Results: As is shown in Table V, MFG Augment substantially improves the performance over the three baseline augmentation methods. The improvement is especially prominent when training on small ImageNet datasets. When training on 1/4 Imagenet, our MFG Augment can achieve test accuracy at 61.13%, which is +6.06% higher than Cutout, +3.56% higher than Cutmix, and +4.1% higher than Augmix. In addition, When training on full Imagenet, our MFG Augment can achieve the highest test accuracy at 72.26%.

Model	ResNet-18	ResNet-50
Baseline	69.76	76.15
Cutout	70.25	77.01
Cutmix	71.52	78.41
Augmix	71.49	77.68
MFG Augment	**72.26**	**79.22**

TABLE VI
TOP-1 TEST ACCURACY (%) ON IMAGENET.

When trained on ResNet-50, our MFG Augment again achieves the highest test accuracy. As is shown in Table VI, on ResNet-18, our MFG Augment achieves 72.26% test accuracy, which is +2.01% higher than Cutout, +0.74% higher than Cutmix, +0.77% higher than Augmix. On ResNet-50, our MFG Augment achieves 79.22% test accuracy, which is +2.21% higher than Cutout, +0.81% higher than Cutmix, +1.54% higher than Augmix.

E. Affinity and Diversity Analysis

In this subsection, we compare our MFG Augment with the state-of-art augmentation methods: Cutout,Cutmix and Augmix, in terms of "Affinity" and "Diversity", proposed in [99]. The experiment is trained with EfficientNet on CIFAR-10 database. All the hyperparameter settings are the same as in the experiments of CIFAR-10 in Section V-C. According to [99], test accuracy usually improves when moving to the upper right region in the "Affinity-Diversity plane". We can observe in Fig. 6 that our MFG Augment is in the upper right corner of this plane, which means that it has the highest affinity and diversity among all the comparative augmentation methods. Specifically, the affinity of our Augment is about 0.92, and the diversity is about 2.5, while for Cutout, Cutmix and Augmix, all have affinity < 0.72 and diversity < 2.1. There is an outlier for Cutout and Cutmix, respectively. The results serve as an explanation for why MFG Augment is more effective than Cutout, Cutmix and Augmix.

VI. CONCLUSION

In this paper, we propose MFG Augment, a data augmentation based on the Mean-field Game theory that can generate a "path" in images' pixel space and feature space, and the points along the path are augmented images' pixels or features. MFGs introduce a systematic and theoretically grounded approach when applied to data augmentation. By representing the collective behavior of individual agents (i.e,

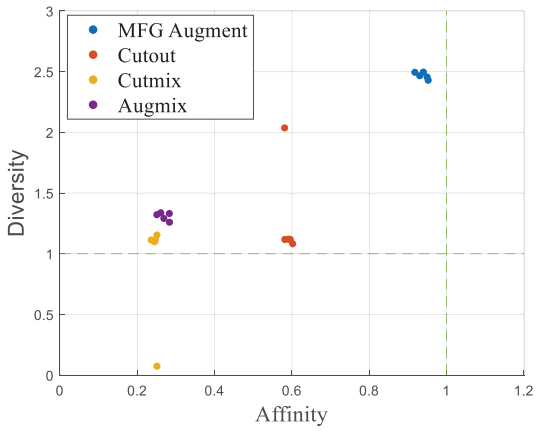


Fig. 6. Affinity and diversity plane for Cutout, Cutmix, Augmix, and our MFG Augment on CIFAR-10, trained with EfficientNet.

image pixels and learned features in the proposed `MFG Augment`), it provides a mathematical framework to generate and analyze new data that's consistent, relevant, and theoretically validated. We demonstrate that our `MFG Augment` achieves state-of-the-art test accuracy on MNIST, CIFAR-10, and ImageNet. The high performance is explained by the increased diversity and affinity of the augmented data. Importantly, `MFG Augment` generates impressive results for small datasets. However, the time complexity of the proposed `MFG Augment` can be high for large datasets. Also, another concern is ensuring the augmented data remains true to real-world scenarios. Looking ahead, there is potential to explore how to reduce the computation complexity in `MFG Augment` for larger datasets. Additionally, combining `MFG Augment` with existing data augmentation techniques to further improve the model performance may also be considered in our future work.

REFERENCES

[1] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding data augmentation for classification: when to warp?" in *Int. Conf. Digital Image Comput. Tech. Appl. (DICTA)*, Gold Coast, Australia, Nov. 2016.

[2] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," in *2018 IEEE symposium series on computational intelligence (SSCI)*, Bangalore, India, Nov. 2018, pp. 1542–1547.

[3] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, Aug. 2017.

[4] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501*, May. 2018.

[5] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2020.

[6] C. Gong, D. Wang, M. Li, V. Chandra, and Q. Liu, "Keepaugment: A simple information-preserving data augmentation approach," in *Proc. IEEE/CVF conf. Comput. Vis. Patt. Recog.*, Jun. 2021.

[7] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, "Improved noisy student training for automatic speech recognition," *arXiv preprint arXiv:2005.09629*, May. 2020.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, Oct. 2018.

[9] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *2012 IEEE Conf. Comput. Vis. Patt. Recog.*, Providence, RI, Jun. 2012.

[10] I. Sato, H. Nishimura, and K. Yokoi, "Apac: Augmented pattern classification with neural networks," *arXiv preprint arXiv:1505.03229*, May. 2015.

[11] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019.

[12] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," *arXiv preprint arXiv:1912.02781*, Oct. 2019.

[13] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *Int. Conf. Machine Learning*, Long Beach, California, Jun. 2019.

[14] T. DeVries and G. W. Taylor, "Dataset augmentation in feature space," *arXiv preprint arXiv:1702.05538*, Feb. 2017.

[15] B. Liu, X. Wang, M. Dixit, R. Kwitt, and N. Vasconcelos, "Feature space transfer for data augmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, Utah, Jun. 2018, pp. 9090–9098.

[16] H. Nishizaki, "Data augmentation and feature extraction using variational autoencoder for acoustic modeling," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit Conf. (APSIPA ASC)*, Kuala Lumpur, Malaysia, Dec. 2017.

[17] J.-M. Lasry and P.-L. Lions, "Mean field games," *Japanese journal of mathematics*, vol. 2, no. 1, pp. 229–260, Mar. 2007.

[18] P. E. Caines, "Mean field games," in *Encyclopedia of Systems and Control*, 2021, pp. 1197–1202.

[19] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A comprehensive survey of image augmentation techniques for deep learning," *Pattern Recognition*, p. 109347, May 2023.

[20] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Inte. Conf. Machine Learning*, Sydney, Australia, Aug. 2017.

[21] R. Flamary, M. Cuturi, N. Courty, and A. Rakotomamonjy, "Wasserstein discriminant analysis," *Machine Learning*, vol. 107, no. 12, pp. 1923–1945, May. 2018.

[22] F.-P. Paty and M. Cuturi, "Subspace robust wasserstein distances," in *Inte. Conf. Machine Learning*, Long Beach, California, Jun. 2019.

[23] S. Ferradans, N. Papadakis, G. Peyré, and J.-F. Aujol, "Regularized discrete optimal transport," *SIAM J. Imaging Sciences*, vol. 7, no. 3, pp. 1853–1882, 2014.

[24] D. Alvarez-Melis and T. S. Jaakkola, "Gromov-wasserstein alignment of word embedding spaces," *arXiv preprint arXiv:1809.00013*, Aug. 2018.

[25] Y. Kang, S. Liu, H. Zhang, W. Li, Z. Han, S. Osher, and H. V. Poor, "Joint sensing task assignment and collision-free trajectory optimization for mobile vehicle networks using mean-field games," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8488–8503, Dec. 2020.

[26] H. Zhang, Y. Kang, L. Song, Z. Han, and H. V. Poor, "Age of information minimization for grant-free non-orthogonal massive access using mean-field games," *IEEE Trans Commun.*, vol. 69, no. 11, pp. 7806–7820, Aug. 2021.

[27] L. Li, Q. Cheng, X. Tang, T. Bai, W. Chen, Z. Ding, and Z. Han, "Resource allocation for noma-mec systems in ultra-dense networks: A learning aided mean-field game approach," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 3, pp. 1487–1500, Nov. 2020.

[28] Y. Kang, H. Wang, B. Kim, J. Xie, X.-P. Zhang, and Z. Han, "Time efficient offloading optimization in automotive multi-access edge computing networks using mean-field games," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6460–6473, May. 2023.

[29] P. Y. Simard, D. Steinkraus, J. C. Platt *et al.*, "Best practices for convolutional neural networks applied to visual document analysis," vol. 3, no. 2003, Aug. 2003.

[30] K. K. Singh, H. Yu, A. Sarmasi, G. Pradeep, and Y. J. Lee, "Hide-and-seek: A data augmentation technique for weakly-supervised localization and beyond," *arXiv preprint arXiv:1811.02545*, Nov. 2018.

[31] P. Chen, S. Liu, H. Zhao, and J. Jia, "Gridmask data augmentation," *arXiv preprint arXiv:2001.04086*, Jan. 2020.

[32] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 07, Apr., pp. 13 001–13 008.

[33] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, Oct. 2017.

[34] H. Inoue, "Data augmentation by pairing samples for images classification," *arXiv preprint arXiv:1801.02929*, Jan. 2018.

[35] Y. Tokozume, Y. Ushiku, and T. Harada, "Between-class learning for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, USA, Jun. 2018, pp. 5486–5494.

[36] J.-H. Kim, W. Choo, H. Jeong, and H. O. Song, "Co-mixup: Saliency guided joint mixup with supermodular diversity," *arXiv preprint arXiv:2102.03065*, Feb. 2021.

[37] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017.

[38] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Nashville, USA, Jun. 2021.

[39] Z. Xu, A. Meng, Z. Shi, W. Yang, Z. Chen, and L. Huang, "Continuous copy-paste for one-stage multi-object tracking and segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Montreal, Canada, Oct. 2021.

[40] E. Harris, A. Marcu, M. Painter, M. Niranjan, A. Prügel-Bennett, and J. Hare, "Fmix: Enhancing mixed sample data augmentation," *arXiv preprint arXiv:2002.12047*, Feb. 2020.

[41] A. Dabouei, S. Soleymani, F. Taherkhani, and N. M. Nasrabadi, "Supermix: Supervising the mixing data augmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 13 794–13 803.

[42] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka, "Synthesizing training data for object detection in indoor scenes," *arXiv preprint arXiv:1702.07836*, Feb. 2017.

[43] N. Dvornik, J. Mairal, and C. Schmid, "Modeling visual context is key to augmenting object detection datasets," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, Sept. 2018.

[44] V. Olsson, W. Tranheden, J. Pinto, and L. Svensson, "Classmix: Segmentation-based data augmentation for semi-supervised learning," in *IEEE Winter Conf. Appl. Comput. Vis.*, Waikoloa, HI, Jan. 2021.

[45] C. Summers and M. J. Dinneen, "Improved mixed-example data augmentation," in *2019 IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Waikoloa, HI, Jan. 2019.

[46] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," *arXiv preprint arXiv:1711.04340*, Nov. 2017.

[47] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using gan for improved liver lesion classification," in *IEEE Int Symp Biomed (ISBI 2018)*, Washington, DC, Apr. 2018.

[48] F. H. K. D. S. Tanaka and C. Aranha, "Data augmentation using gans," *arXiv preprint arXiv:1904.09135*, Apr. 2019.

[49] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "Bagan: Data augmentation with balancing gan," *arXiv preprint arXiv:1803.09655*, Mar. 2018.

[50] V. Sandfort, K. Yan, P. J. Pickhardt, and R. M. Summers, "Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks," *Scientific reports*, vol. 9, no. 1, p. 16884, Nov. 2019.

[51] T. Tran, T. Pham, G. Carneiro, L. Palmer, and I. Reid, "A bayesian data augmentation approach for learning deep models," *Advances in neural information processing systems*, vol. 30, Dec. 2017.

[52] A. Ali-Gombe and E. Elyan, "Mfc-gan: Class-imbalanced dataset classification using multiple fake class generative adversarial network," *Neurocomputing*, vol. 361, pp. 212–221, Oct. 2019.

[53] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Syst. Appl.*, vol. 91, pp. 464–471, Jan 2018.

[54] S.-W. Huang, C.-T. Lin, S.-P. Chen, Y.-Y. Wu, P.-H. Hsu, and S.-H. Lai, "Auggan: Cross domain adaptation with gan-based data augmentation," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, Sept. 2018.

[55] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *Int. Interdisciplinary PhD Workshop (IIPhDW)*, Poland, May 2018, pp. 117–122.

[56] P. T. Jackson, A. A. Abarghouei, S. Bonner, T. P. Breckon, and B. Obara, "Style augmentation: data augmentation via style randomization." in *CVPR workshops*, Long Beach, USA, Jun. 2019.

[57] X. Zhang, Q. Wang, J. Zhang, and Z. Zhong, "Adversarial autoaugment," *arXiv preprint arXiv:1912.11188*, Dec. 2019.

[58] D. Ho, E. Liang, X. Chen, I. Stoica, and P. Abbeel, "Population based augmentation: Efficient learning of augmentation policy schedules," in *Int. Conf. Mach. Learn.*, Long Beach, USA, Jun. 2019.

[59] P. Li, X. Liu, and X. Xie, "Learning sample-specific policies for sequential image augmentation," in *Proc. ACM Int. Conf. Multimedia*, Nice, France, Oct. 2021.

[60] Z. Tang, X. Peng, T. Li, Y. Zhu, and D. N. Metaxas, "Adatransform: Adaptive data transformation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, Korea, Nov. 2019.

[61] A. J. Ratner, H. Ehrenberg, Z. Hussain, J. Dunnmon, and C. Ré, "Learning to compose domain-specific transformations for data augmentation," *Adv. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017.

[62] D. Lee, H. Park, T. Pham, and C. D. Yoo, "Learning augmentation network via influence functions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, USA, Jun. 2020.

[63] T. Takase, R. Karakida, and H. Asoh, "Self-paced data augmentation for training neural networks," *Neurocomputing*, vol. 442, pp. 296–306, Jun. 2021.

[64] J.-H. Kim, W. Choo, and H. O. Song, "Puzzle mix: Exploiting saliency and local statistics for optimal mixup," in *Int. Conf. Machine Learning*, Vienna, Austria, Jul. 2020.

[65] B. Li, F. Wu, S.-N. Lim, S. Belongie, and K. Q. Weinberger, "On feature normalization and data augmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, Jun. 2021.

[66] C.-W. Kuo, C.-Y. Ma, J.-B. Huang, and Z. Kira, "Featmatch: Feature-based augmentation for semi-supervised learning," in *Eur. Conf. Comput. Vis.*, Glasgow, UK, Aug. 2020.

[67] P. Li, D. Li, W. Li, S. Gong, Y. Fu, and T. M. Hospedales, "A simple feature augmentation for domain generalization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, Canada, Oct. 2021.

[68] R. Volpi, P. Morerio, S. Savarese, and V. Murino, "Adversarial feature augmentation for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, USA, Jun. 2018, pp. 5495–5504.

[69] Z. Chen, Y. Fu, Y. Zhang, Y.-G. Jiang, X. Xue, and L. Sigal, "Multi-level semantic feature augmentation for one-shot learning," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4594–4605, Sept. 2019.

[70] J. Lee, E. Kim, J. Lee, J. Lee, and J. Choo, "Learning debiased representation via disentangled feature augmentation," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 25 123–25 133, Oct. 2021.

[71] Y.-C. Chen, X. Zhu, W.-S. Zheng, and J.-H. Lai, "Person re-identification by camera correlation aware feature augmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 392–408, Feb. 2018.

[72] Y. Zhang, H. Zhu, Z. Song, P. Koniusz, and I. King, "Spectral feature augmentation for graph contrastive learning and beyond," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 9, Washington DC, USA, Feb. 2023, pp. 11 289–11 297.

[73] T. Chen, Y. Cheng, Z. Gan, J. Wang, L. Wang, Z. Wang, and J. Liu, "Adversarial feature augmentation and normalization for visual recognition," *arXiv preprint arXiv:2103.12171*, Mar. 2021.

[74] Z. Gao, Y. Wu, Y. Jia, and M. Harandi, "Hyperbolic feature augmentation via distribution estimation and infinite sampling on manifolds," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 34 421–34 435, Nov. 2022.

[75] Y. Wang, X. Pan, S. Song, H. Zhang, G. Huang, and C. Wu, "Implicit semantic data augmentation for deep networks," *Adv. Neural Inf. Process. Syst.*, Dec. 2019.

[76] Y. Wang, G. Huang, S. Song, X. Pan, Y. Xia, and C. Wu, "Regularizing deep networks with semantic data augmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3733–3748, Jul. 2021.

[77] H. Li, X. Zhang, Q. Tian, and H. Xiong, "Attribute mix: Semantic data augmentation for fine grained recognition," in *IEEE Int. Conf. Vis. Commun. Image Process.*, Macau, China, Dec. 2020.

[78] J. Zhang, Y. Zhang, and X. Xu, "Objectaug: object-level data augmentation for semantic image segmentation," in *Int. Jt. Conf. Neural Netw.*, Shenzhen, China, Jul. 2021.

[79] X. Chen, Y. Zhou, D. Wu, W. Zhang, Y. Zhou, B. Li, and W. Wang, "Imagine by reasoning: A reasoning-based implicit semantic data augmentation for long-tailed classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 1, Feb. 2022, pp. 356–364.

[80] Y. Zang, C. Huang, and C. C. Loy, "Fasa: Feature augmentation and sampling adaptation for long-tailed instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, Canada, Mar. 2021.

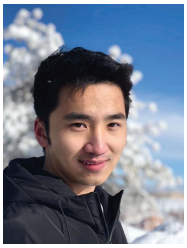
[81] J. Liu, C. Chai, Y. Luo, Y. Lou, J. Feng, and N. Tang, "Feature augmentation with reinforcement learning," in *IEEE Int. Conf. Data Eng.*, Kuala Lumpur, Malaysia, May 2022.

[82] M. Jacobs, F. Léger, W. Li, and S. Osher, "Solving large-scale optimization problems with a convergence rate independent of grid size," *SIAM J. Numer.l Analy.*, vol. 57, no. 3, pp. 1100–1123, 2019.

[83] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai, "Better mixing via deep representations," in *Int. Conf. Machine Learning*, Atlanta, GA, Jun. 2013.

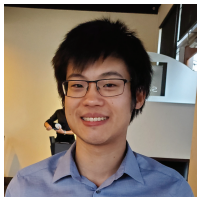
[84] S. Ozair and Y. Bengio, "Deep directed generative autoencoders," *arXiv preprint arXiv:1410.0630*, Oct. 2014.

[85] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, Dec. 2013.
[86] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in Neural Inform. Processing Syst.*, Barcelona, Spain, Dec. 2016.
[87] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, Jul. 1966.
[88] J. Schulte, "Adjoint methods for hamilton-jacobi-bellman equations," *Diploma Thesis*, 2010.
[89] A. T. Lin, S. W. Fung, W. Li, L. Nurbekyan, and S. J. Osher, "Apac-net: Alternating the population and agent control via two neural networks to solve high-dimensional stochastic mean field games," *arXiv preprint arXiv:2002.10113*, Feb. 2020.
[90] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 27, Dec. 2014.
[91] L. Weng, "From gan to wgan," *arXiv preprint arXiv:1904.08994*, Apr. 2019.
[92] S.-i. Amari, R. Karakida, and M. Oizumi, "Information geometry connecting wasserstein distance and kullback–leibler divergence via the entropy-relaxed transportation problem," *Inf. Geom.*, vol. 1, pp. 13–37, Mar. 2018.
[93] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[94] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," in *Citeseer*, Toronto, Canada, Apr. 2009.
[95] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conf. Comput. vis. Patt. Recog.*, Miami, FL, Jun. 2009.
[96] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Inte. Conf. Machine Learning*, Long Beach, California, Jun. 2019.
[97] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. Comput. Vis. Patt. Recog.*, Las Vegas, Nevada, Jun. 2016.
[98] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, Aug. 2016.
[99] R. Gontijo-Lopes, S. J. Smullin, E. D. Cubuk, and E. Dyer, "Affinity and diversity: Quantifying mechanisms of data augmentation," *arXiv preprint arXiv:2002.08973*, Feb. 2020.



Yuhan Kang (S'19) received the B.S. at the School of Information and Communication Engineering at University of Electronic Science and Technology of China in 2019. Currently, he is a Ph.D. student in the Electrical and Computer Engineering Department at the University of Houston, Texas. His current research interest includes Mean-field game theory, Deep Learning, Machine Learning, Reinforcement Learning, Internet-of-Things networks, Federated Learning, Edge Computing, and Optimization theory.

Samira Zare received a B.Sc. degree in mechanical engineering from the Sharif University of Technology, Tehran, Iran, in 2015 and an M.Sc. degree in mechanical engineering from the University of Tehran, Tehran, Iran, in 2018. She is a Ph.D. candidate in electrical and computer engineering at the University of Houston, Houston, TX. She is the author of six peer-reviewed research articles, and her research interests include out-of-distribution generalization, bias removal, fairness, and the interpretability of deep learning models. She was the reviewer for the conferences Medical Image Computing and Computer-Assisted Intervention (MICCAI) in 2023 and 2024 and the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) in 2024. Ms. Zare was a recipient of the Travel Award from the Medical Image Computing and Computer-Assisted Intervention (MICCAI) in 2022.

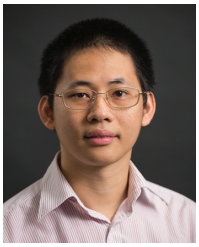


Alex Tong Lin received his B.S in Mathematics from UCSB (University of California, Santa Barbara), Summa Cum Laude, in 2013. He then enrolled in graduate studies in UCLA (University of California, Los Angeles) in 2014 and received his Ph.D. in Mathematics (Applied) in 2020. His work involved the intersection of machine learning/artificial intelligence, and optimal control and multi-agent systems. Today, Alex is a Senior Research Scientist at Discover Financial Services, where he works on unsupervised and semi-supervised learning, and explainability.

Zhu Han (S'01–M'04-SM'09-F'14) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. Dr. Han's main research targets on the novel game-theory related concepts critical to enabling efficient and distributive use of wireless networks with limited resources. His other research interests include wireless resource allocation and management, wireless communications and networking, quantum computing, data science, smart grid, carbon neutralization, security and privacy. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, IEEE Vehicular Technology Society 2022 Best Land Transportation Paper Award, and several best paper awards in IEEE conferences. Dr. Han was an IEEE Communications Society Distinguished Lecturer from 2015 to 2018 and ACM Distinguished Speaker from 2022 to 2025, AAAS fellow since 2019, and ACM Fellow since 2024. Dr. Han is a 1% highly cited researcher since 2017 according to Web of Science. Dr. Han is also the winner of the 2021 IEEE Kiyo Tomiyasu Award (an IEEE Field Award), for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: "for contributions to game theory and distributed management of autonomous communication networks."

Stan Osher received his PhD in Mathematics from New York University in 1966. He has been at UCLA since 1976. He now is a Professor of Mathematics, Computer Science, Electrical Engineering and Chemical and Biomolecular Engineering. He has been elected to $>$ the US National Academy of Science, the US National Academy of Engineering and the American Academy of Arts and Sciences. He was awarded the SIAM Pioneer Prize at the 2003 ICIAM conference and the Ralph E. Kleinman Prize in 2005. He was awarded honorary $>$ doctoral degrees by ENS Cachan, France, in 2006 and by Hong Kong Baptist University in 2009. He is a SIAM and AMS Fellow. He gave a one hour plenary address at the 2010 International Conference of Mathematicians. He also gave the John von Neumann Lecture at the SIAM 2013 annual meeting. He is a Thomson-Reuters/ Clarivate highly cited researcher-among the top 1% from 2002-present in both Mathematics and Computer Science with an h index of 120. In 2014 he received the Carl Friedrich Gauss Prize from the International Mathematics Union-this is regarded as the highest prize in applied mathematics. In 2016 he received the William Benter Prize. His current interests involve data science, which includes optimization, image processing, compressed sensing, machine learning, neural nets and applications of these techniques.



Hien Van Nguyen is an Associate Professor of Computer Engineering at the University of Houston (UH), Texas. He earned his Ph.D. from the University of Maryland College Park in 2013 and a Bachelor's degree in Electrical and Computer Engineering from the National University of Singapore in 2007. Before joining UH, Dr. Nguyen was a Senior Scientist at Uber Advanced Technology Group in Pittsburgh and a Scientist at Siemens Corporate Research in Princeton.

Dr. Nguyen has published 5 technical books and over 70 peer-reviewed publications. He holds 12 U.S. patents and is a senior member of the National Academy of Inventors. His research interests lie at the nexus of machine learning and medicine. His contributions to the field have been recognized with numerous awards and honors, including Best Poster Awards at the International Renal Pathology Conference.

As the Director of the Houston Learning Algorithm Lab, Dr. Nguyen has played a pivotal role in securing over $12 million in research grants from the National Institutes of Health (NIH) and the National Science Foundation (NSF). He currently serves as an Associate Editor of the Computerized Medical Imaging and Graphics journal.