First-Order Manifold Data Augmentation for Regression Learning

Ilya Kaufman [1] **Omri Azencot** [1]

Abstract

Data augmentation (DA) methods tailored to specific domains generate synthetic samples by applying transformations that are appropriate for the characteristics of the underlying data domain, such as rotations on images and time warping on time series data. In contrast, *domain-independent* approaches, e.g. `mixup`, are applicable to various data modalities, and as such they are general and versatile. While regularizing classification tasks via DA is a well-explored research topic, the effect of DA on regression problems received less attention. To bridge this gap, we study the problem of domain-independent augmentation for regression, and we introduce `FOMA`: a new data-driven domain-independent data augmentation method. Essentially, our approach samples new examples from the tangent planes of the train distribution. Augmenting data in this way aligns with the network tendency towards capturing the dominant features of its input signals. We evaluate `FOMA` on in-distribution generalization and out-of-distribution robustness benchmarks, and we show that it improves the generalization of several neural architectures. We also find that strong baselines based on `mixup` are less effective in comparison to our approach. Our code is publicly available at https://github.com/azencot-group/FOMA

1. Introduction

Classification and regression problems primarily differ in their output's domain. In classification, we have a finite set of labels, whereas in regression, the range is an infinite set of quantities—either discrete or continuous. In classical work (Devroye et al., 2013), classification is argued to be "easier" than regression, but more generally, it is agreed by many that classification and regression problems should be treated differently (Muthukumar et al., 2021). Particularly, the differences between classification and regression are actively explored in the context of regularization. Regularizing neural networks to improve their performance on new samples has received a lot of attention in the past few years. One of the main reasons for this increased interest is that most of the recent successful neural models are *overparameterized*. Namely, the amount of learnable parameters is significantly larger than the number of available training samples (Allen-Zhu et al., 2019a;b), and thus regularization is often necessary to alleviate overfitting issues. Recent studies on overparameterized linear models identify conditions under which overfitting is "benign" in regression (Bartlett et al., 2020), and uncover the relationship between the choice of loss functions in classification and regression tasks (Muthukumar et al., 2021). Still, the regularization of deep neural regression networks is not well understood.

In this work, we focus on regularizing deep models via Data Augmentation (DA), where data samples are artificially generated and used during training. In general, DA techniques can be categorized into domain-dependent (DD) methods and domain-independent (DI) approaches. The former are specific for a certain data modality such as images or text, whereas the latter typically do not depend on the data format. Numerous DD- and DI-DA approaches are available for classification tasks (Shorten & Khoshgoftaar, 2019; Shorten et al., 2021), and many of them consistently improve over non-augmented models. Unfortunately, DI-DA for regression problems is underexplored. Recent works on linear models study the connection between the DA policy and optimization (Hanin & Sun, 2021), as well as the generalization effects of linear DA transformations (Wu et al., 2020). We contribute to this line of work by proposing and analyzing a new domain-independent data augmentation method for nonlinear deep regression, and by testing our approach on in-distribution generalization and out-of-distribution robustness tasks (Yao et al., 2022).

Many strong data augmentation methods were proposed in the past few years. Particularly relevant to our study is the family of `mixup`-based techniques that are commonly used in classification applications. The original method, `mixup` (Zhang et al., 2017), produces convex combinations of training samples, promoting linear behavior for

[1]Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel. Correspondence to: Ilya Kaufman <ilyakau@post.bgu.ac.il>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

in-between samples. The method is domain-independent and data-agnostic, i.e., it is indifferent to the given data samples. `Mixup` was shown to solve the Vicinal Risk Minimization (VRM) problem instead of the usual Empirical Risk Minimization (ERM) problem. In comparison, our approach can also be viewed as solving a VRM problem, and it is domain-independent and *data-driven*, namely, augmentations depend on the given data distribution. In our extensive evaluations, we will show that `mixup`-based methods are less effective for regression in comparison to our approach.

Contributions. Challenged by the differences between classification and regression and motivated by the success of domain-independent methods such as `mixup`, we propose a simple, domain-independent and data-driven DA routine, termed First-Order Manifold Augmentation (`FOMA`). Let X, Y be the input and output mini-batch tensors, respectively, and let $Z_l = g_l(X)$ be the hidden code at layer l. Our method produces new training samples $Z_l(\lambda), Y(\lambda)$ from the given ones by scaling down their small singular values by a random $\lambda \in [0, 1]$. At its core, `FOMA` incorporates into training the assumption that data with similar dominant components of the train set should be treated as true samples. Our implementation of `FOMA` is fully differentiable, and thus it is applicable to any layer of a given network.

We detail `FOMA` in Sec. 3, motivating our design choices and illustrating its effect on data. We analyze our approach using perturbation theory and introduce its associated vicinal risk minimization (Sec. 4). Our experimental evaluation focuses on in-distribution generalization (Sec. 5.1) and on out-of-distribution robustness (Sec. 5.2), where we empirically demonstrate the superiority of `FOMA`. We offer a potential explanation to the success of our method (Sec. 3, App. B). Finally, an ablation study is performed, justifying our design choices (Sec. 5.3).

2. Related Work

Deep neural networks regularization is an established research topic with several existing works (Goodfellow et al., 2016). Common regularization approaches include weight decay, dropout (Srivastava et al., 2014), batch normalization (Ioffe & Szegedy, 2015), and data augmentation (DA). Here, we categorize DA techniques to be either domain-dependent or domain-independent. Domain-dependent DA was shown to be effective for, e.g., image data (LeCun et al., 1998) and audio signals (Park et al., 2019), among other domains. However, adapting these methods to new data formats is typically challenging and often infeasible. While several works focused on automatic augmentation (Lemley et al., 2017; Cubuk et al., 2019; Lim et al., 2019; Tian et al., 2020; Cubuk et al., 2020), there is concurrently an increased interest on domain-independent DA methods, allowing to regularize neural networks when only basic data assumptions are allowed (Naiman et al., 2023). We focus in what follows on *domain-independent* techniques that were proposed in the context of classification and regression problems.

DA for classification. Zhang et al. (2017) proposed to perform convex mixing of input samples as well as one-hot output labels during training. The new training procedure, named `mixup`, minimizes the Vicinal Risk Minimization (VRM) problem instead of the typical Empirical Risk Minimization (ERM). Many extensions of `mixup` were proposed, including mixing latent features (Verma et al., 2019), same-class mixing (DeVries & Taylor, 2017), among other extensions (Guo et al., 2019; Hendrycks et al., 2019; Yun et al., 2019; Berthelot et al., 2019; Greenewald et al., 2021; Lim et al., 2021). ISDA (Wang et al., 2019) formulates a new cross-entropy loss for DA-based training using the per-class covariance matrix.

DA for regression. Significantly less attention has been drawn to designing domain-independent data augmentation for regression tasks. A recent survey (Wen et al., 2020) on DA for time series data lists a few basic augmentation tools. Dubost et al. (2019) propose to recombine samples for regression tasks with countable outputs, and thus their method can not be directly extended to the uncountable regime. `RegMix` (Hwang & Whang, 2021) developed a meta learning framework based on reinforcement learning for mixing samples in their neighborhood. A recent work (Yao et al., 2022) showed that applying vanilla mixup with adjusted sampling probabilities based on label similarity can improve generalization on regression tasks. Another work (Schneider et al., 2023) suggests DA based on Anchor regression (Rothenhäusler et al., 2021) which allows mixing multiple samples based on their cluster that encodes a homogeneous group of observations.

3. First-Order Manifold Augmentation

A learning task is typically described as a function which maps inputs to outputs. In this view, a learning model is approximating that function using e.g., a neural network, and it is formulated via $f : \mathcal{X} \to \mathcal{Y}$, denoting the input and output domains by $\mathcal{X}$ and $\mathcal{Y}$, respectively. A regression problem is such that the output domain is (un)countable, e.g., $\mathcal{Y} \subset \mathbb{N}^m$ or $\mathcal{Y} \subset \mathbb{R}^m$. We consider the general setting where $\mathcal{X} \subset \mathbb{R}^n, \mathcal{Y} \subset \mathbb{R}^m$. During training, the learning model is provided with a training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, sampled from $(x_i, y_i) \sim \mathcal{P}$. Our method extends $\mathcal{D}$ by producing a new training distribution as we describe below.

To generate new samples, we consider the singular value decomposition (`SVD`) of a matrix $M \in \mathbb{R}^{q \times r}, q \geq r$ which is given by $M = USV^T$. U, V are orthogonal, and S

```
def scale(A, k, lam):
  U, s, Vt = torch.linalg.svd(A, full_matrices=False)
  lam_repeat = lam.repeat(s.shape[-1] - k)
  lam_ones = torch.ones(k)
  s = s * torch.cat((lam_ones, lam_repeat))
  A = U @ torch.diag(s) @ Vt
  return A

# X, Y are in batch x feats
for (X, Y) in loader:
  lam = beta.Beta(alpha, alpha).sample()
  A = torch.concatenate((X, Y), axis=1)
  A = scale(A, k, lam)
  X, Y = A[:, :n], A[:, n:]
  optimizer.zero_grad()
  loss(net(X), Y).backward()
  optimizer.step()
```

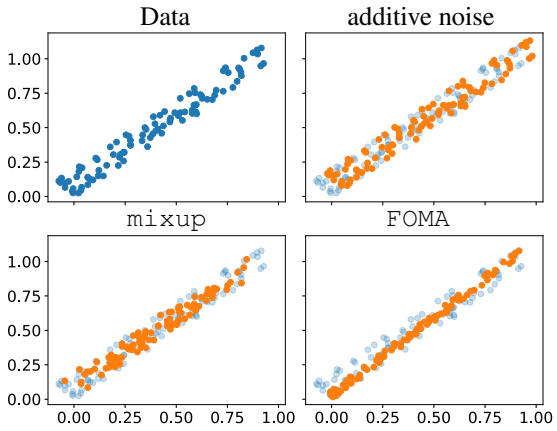

Figure 1. We show the pseudocode for FOMA at the input level, $l = 0$ (left). We demonstrate the effect of a few DA methods on 2D data whose intrinsic dimension is one (right).

is diagonal consisting of the singular values ordered by $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$. SVD is intimately related to principal component analysis (PCA) which in turn is heavily studied in manifold learning and dimensionality reduction (Ma & Fu, 2012). It is well known that the best rank k approximation of M is given by omitting the last $(r-k)$ singular values, i.e., $M_k = \sum_{j=1}^{k} \sigma_j u_j v_j^T$ (Eckart & Young, 1936). The matrix M_k preserves the (k) dominant components in M, and discards the rest. Further, SVD is also known to yield a first-order approximation of the data manifold (Singer & Wu, 2012; Kaufman & Azencot, 2023). Our key insight is that scaling the small singular values produces training samples that are in close proximity to the manifold, and thus to the true data distribution $\mathcal{P}$.

In particular, based on the manifold hypothesis (Cayton et al., 2008), we assume that the data samples $\mathcal{D}$ live on or close to a manifold $\mathcal{M} \subset \mathcal{X} \times \mathcal{Y}$. We denote by $\mathcal{T}(x, y)$ the tangent plane of the data manifold $\mathcal{M}$ at the point $(x, y) \in \mathcal{M}$. Namely, $\mathcal{T}(x, y)$ is the linear approximation of $\mathcal{M}$ at (x, y). Below, we utilize SVD to approximate $\mathcal{T}(x, y)$, and to generate artificial samples by considering pairs $(\tilde{x}, \tilde{y})$ in the tangent plane of the (x, y).

Let the input and output mini-batch tensors $X \in \mathbb{R}^{b \times n_0}$ and $Y \in \mathbb{R}^{b \times m}$, respectively, where w.l.o.g $b \geq n_0 + m$ is the batch size. We denote the network by $f(X) = f_l(g_l(X))$, $Z_l := g_l(X)$ where g_l maps inputs to latent representations $Z_l \in \mathbb{R}^{b \times n_l}$ at layer $l \in [0, L]$, and f_l maps latent vectors to outputs. Let $\lambda \sim \text{Beta}(\alpha, \alpha)$ for $\alpha \in (0, \infty)$ and $k \in [1, n_l + m]$ be the index of the singular value after which we scale down. Then, the new artificial samples $Z_l(\lambda, k), Y(\lambda, k)$ are defined via

$$
\begin{aligned}
A &:= [Z_l,\ Y] = USV^T \in \mathbb{R}^{b \times (n_l + m)}\ , \\
A(\lambda,\ k) &:= US(\lambda,\ k)V^T \in \mathbb{R}^{b \times (n_l + m)}\ , \\
Z_l(\lambda,\ k) &:= A(\lambda,\ k)_{1:n_l} \in \mathbb{R}^{b \times n_l}\ , \\
Y(\lambda,\ k) &:= A(\lambda,\ k)_{n_l+1:n_l+m} \in \mathbb{R}^{b \times m}\ ,
\end{aligned}
$$

where $[\cdot, \cdot]$ concatenates along columns, $A_{1:i}$ stands for the first i column vectors in A, and $S(\lambda,\ k)$ is the diagonal matrix of scaled down singular values. Namely, we compute $S(\lambda,\ k) = \text{diag}(\sigma_1, \ldots, \sigma_k \mid \lambda\sigma_{k+1}, \ldots, \lambda\sigma_{n_l+m})$. We propose two methods for choosing the parameter k, one is based on the intrinsic dimension of the data, denoted by FOMA and the second is based on the explained variance of the data denoted by FOMA_ρ.

Intrinsic dimension. We set the value of k to be equal the intrinsic dimension (ID) of A, i.e., the minimal number of features needed to represent the data with little information loss (Facco et al., 2017). The ID can be estimated for the entire dataset at the input level, or alternatively, approximated per batch during the training process. Setting k to equal the ID is motivated by our analysis below in Sec. 4, where we show that FOMA is equivalent to sampling from the tangent space of the manifold. In practice, we use k dominant singular vectors to approximate the tangent space.

Explained variance. In addition to the ID, we also consider the value k to depend on the hyper-parameter $\rho \in [0, 1]$ that represents the "amount" of signal to keep unchanged, i.e.,

$$
k = \arg\max_{\tilde{k}} \sum_{j=1}^{\tilde{k}} \sigma_j / \sum_j \sigma_j \leq \rho\ .
$$

Similar to mixup (Zhang et al., 2017), our method recovers the original dataset $\mathcal{D}$ as $\alpha \to 0, \forall k$.

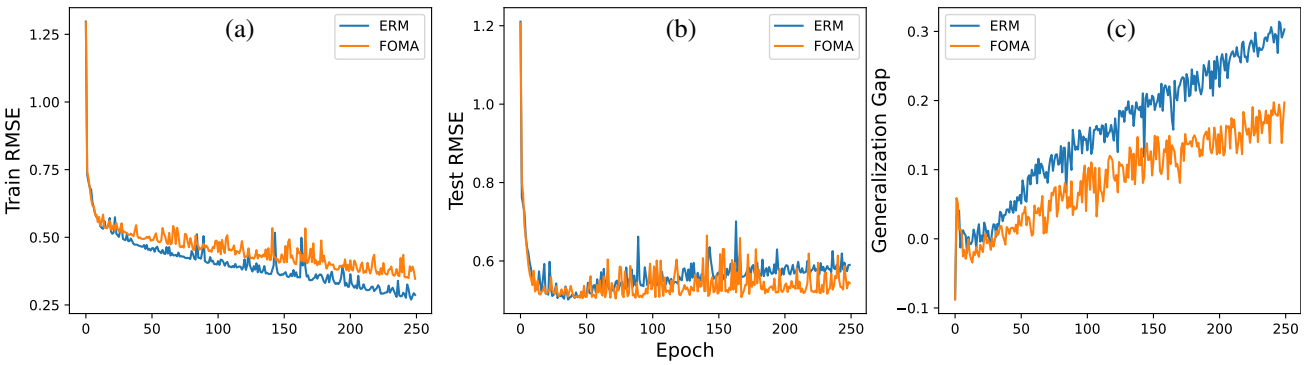

Figure 2. Training stability and overfitting. (a) RMSE loss on the train set. (b) RMSE loss on the test set. (c) Generalization gap: the difference between test error and train error

The loss function associated with FOMA is

$$\mathcal{L}(f) = \mathbb{E}_{(X,Y)}\, \mathbb{E}_{\lambda}\, \mathbb{E}_{l}\, c_{\lambda}\left[(f_l, 1) \circ \chi(g_l(X), Y))\right] \,,$$
$$\text{s.t.} \quad (X,Y) \sim \mathcal{D},\ \lambda \sim \mathcal{H}(\sigma),\ l \sim [0, L] \,,$$

where $c_\lambda : \mathbb{R}^{b\times m} \times \mathbb{R}^{b\times m} \to [0, \infty)$ is a cost function, typically mean squared error (MSE). The `scale` transform χ takes a pair of tensors $g_l(X), Y$, and it scales down by λ the last $(n_l + m - k)$ singular values of their concatenation. A key attribute of FOMA is that it is fully *differentiable* since the singular value decomposition can be back-propagated (Ionescu et al., 2015). We provide an example PyTorch pseudocode in Fig. 1 (left). The computational complexity of FOMA is governed by SVD calculation which has a complexity of $\mathcal{O}(\min(qr^2, rq^2))$ for a $q \times r$ matrix.

Design choices. For certain λ values, the new sample $Z_l(\lambda, k), Y(\lambda, k)$ may be too far from $\mathcal{P}$. With this in mind, we explored the option of scaling down the loss function $c(\cdot, \cdot)$ by a parameter $\mu(\lambda)$ in addition to modifying the singular values. However, we tested various profiles $\mu(\lambda)$ and discovered the most consistent models are obtained when no scaling of loss occurs, see Sec. 5.3. Importantly, this means that our approach adopts a different ansatz in comparison to `mixup`-based methodologies. While `mixup` incorporates uncertainty into the model training using "in-between" samples and labels, our method uses the new data as if it was sampled from the true distribution, since we do not scale c. An alternative option which would be conceptually closer to `mixup` is to scale the *large* singular values as well as the loss term. We show in Sec. 5.3 that this choice is usually inferior to FOMA.

Batch selection. Given a sample $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we aim to produce training samples that are in close proximity to the manifold $\mathcal{M}$, and thus to the true data distribution $\mathcal{P}$. However, we do not know the structure of $\mathcal{M}$, and thus we can not sample from it directly, unfortunately. To overcome this challenge, we sample from a linear approximation of the $\mathcal{M}$ at (x, y) given by the tangent plane $\mathcal{T}(x, y)$. In practice, we need a set of points $N_{(x,y)} = \{(x_i, y_i)\}$ in the neighborhood of (x, y) to estimate $\mathcal{T}(x, y)$ via SVD. The proximity of the points $N_{(x,y)}$ to (x, y) has a direct effect on the quality of the approximation of the tangent plane. In practice, we construct batches of points that are close to each other as measured by the Euclidean distances between the labels in $\mathcal{Y}$. We hypothesize that better approximations should generate samples closer to the data manifold, improving empirical test results. To validate our claim, we tested two batch selection methods for training 1) randomly selecting batches (random); and 2) constructing batches using samples of points that are close to each other (close). For each setting we trained several models and selected the model that achieved the best performance on the validation set. In Table 1, we report the results of batch selection methods on the test set. It is notable that constructing batches of close points outperforms random batch selection.

Table 1. Results for different batch selection methods FOMA.

Dataset	Airfoil ↓	NO2 ↓	Exchange ↓	Electricity ↓	Echo ↓	RCF ↓	Crimes ↓	Poverty ↑	DTI ↑
FOMA- random	2.159	**0.515**	0.014	0.059	5.248	0.175	0.144	0.433	0.491
FOMA- close	**1.646**	0.521	**0.013**	**0.058**	**5.215**	**0.171**	**0.132**	**0.492**	**0.496**
FOMAρ - random	2.012	0.514	0.014	0.060	**5.224**	0.163	0.134	0.409	**0.508**
FOMAρ - close	**1.471**	**0.512**	**0.013**	**0.059**	5.512	**0.159**	**0.128**	**0.488**	0.503

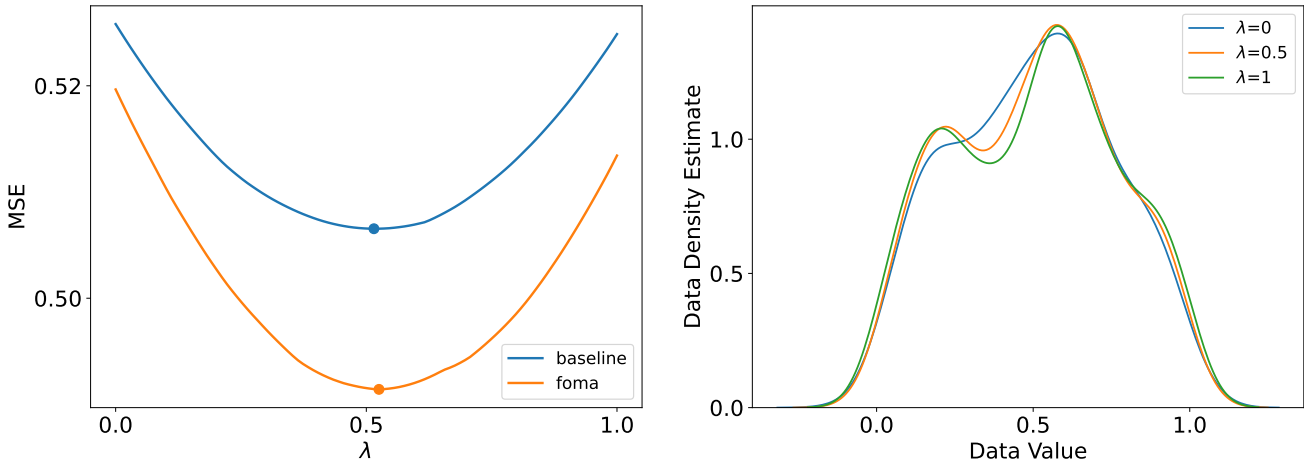

Figure 3. Evaluating a non-augmented model and a model trained with FOMA on train data whose small singular values are scaled down for different values of λ (left). We show on the right panel the probability density function of the original data (green), and its modifications using $\lambda = 0$ (blue), and $\lambda = 0.5$ (orange).

Stability and overfitting We trained two models on the NO2 dataset, one without using our method (ERM) and one with our method (FOMA) and report the results in Fig. 2. We observe that (a) the training process of FOMA is stable with respect to ERM, i.e., the loss is monotonically decreasing with relatively small fluctuations. Furthermore, FOMA exhibits better test performance (b) and smaller generalization gap (c).

Computational resources The computational complexity of FOMA is governed by SVD calculation which has a complexity of $\mathcal{O}(\min(qr^2, rq^2))$ for a $q \times r$ matrix. In Table 2, we compare the average epoch time across 50 epochs in seconds to provide an estimate for the empirical computational cost of FOMA. The results are obtained with a single RTX3090 GPU. Note that the runtime of FOMA is very dependant on the batch size. Larger batch size results in less SVD computations, and depending on implementation, SVD of larger matrices can be evaluated faster.

Table 2. Training times using FOMA (in seconds).

DATASET	AIRFOIL	ELECTRICITY	DTI
ERM	0.058	1.406	33.374
FOMA- INPUT	0.148	3.101	43.433
FOMA- LATENT	0.286	2.837	51.399
FOMA- BOTH	0.321	4.376	63.071
FOMAρ - INPUT	0.119	2.806	41.527
FOMAρ - LATENT	0.149	3.021	43.985
FOMAρ - BOTH	0.262	4.315	57.35

The effect of FOMA on data and learning. We generated a 2D point cloud whose intrinsic dimension is one (shown in blue, Fig. 1), and we applied different DA methods on this data. The three panels in the figure show in orange the augmented data when using additive noise, mixup, and FOMA with $\lambda = 0.5$ and over the original point cloud colored in light blue. Injecting noise alters each point in its neighborhood, whereas mixup draws the points towards the center of their convex hull. In contrast, FOMA aligns the new samples along the dominant component of the original data. Notably, our approach may increase the span of training data, and thus it can improve estimation in regression as was recently shown in (Wu et al., 2020).

We argue that training on samples created with our method encourages the inherent tendency of the network to model the dominant parts of the data better (Naiman & Azencot, 2023). To demonstrate this phenomenon, we trained a three-layer fully connected network with and without FOMA on the NO2 dataset. The trained models are evaluated on the test dataset modified using a 100 varying $\lambda \in [0, 1]$ values, see Fig. 3 (left). Namely, we modify the singular values of every batch in the dataset for each λ, and feed the resulting data for inference. Surprisingly, the non-augmented model (blue curve) performs *better* on the unseen modified samples, yielding the minimum at $\lambda \approx 0.5$. Further, we note that for the majority of λ values, the test MSE is lower than the error obtained for the original data $\mathcal{D}$ (i.e., for $\lambda = 1$). In comparison, the regularized network attains a qualitatively similar plot in terms of the minimizing λ and test MSE profile, however, the MSE is lower for all λ. This example shows that the (non-augmented and augmented) models generalize better to data projected to the manifold $\mathcal{M}$, except for a few low λ values. We conclude that deep regression models may benefit from altering their training procedure by using samples closer to the data manifold $\mathcal{M}$. Finally, this behavior was found to be consistent across several architectures and datasets, see App. B.

Table 3. Comparison of in-distribution generalization tasks. Bold values represent the best results and underlined values are second best. We report the average RMSE and MAPE over three seeds. Full results with standard deviation can be found in App F.

	Airfoil		NO2		Exchange-Rate		Electricity		Echo	
	RMSE ↓	MAPE (%)↓	RMSE ↓	MAPE (%)↓	RMSE ↓	MAPE (%)↓	RMSE ↓	MAPE (%)↓	RMSE ↓	MAPE (%)↓
ERM†	2.901	1.753	0.537	13.615	0.0236	2.423	0.0581	13.861	5.402	<u>8.700</u>
Mixup†	3.730	2.327	0.528	13.534	0.0239	2.441	0.0585	14.306	5.393	8.838
Mani Mixup†	3.063	1.842	0.522	13.382	0.0242	2.475	0.0583	14.556	5.482	8.955
C-Mixup⋆	2.748	1.645	0.516	<u>13.069</u>	0.024	2.456	**0.057**	**13.349**	<u>5.362</u>	8.868
ADA⋆	2.357	1.377	<u>0.515</u>	13.128	0.022	2.250	0.059	13.58	-	-
FOMA	<u>1.646</u>	<u>0.963</u>	0.521	13.23	<u>0.013</u>	<u>1.280</u>	<u>0.058</u>	14.614	**5.215**	**8.331**
FOMA$_\rho$	**1.471**	**0.816**	**0.512**	**12.894**	**0.013**	**1.262**	0.059	<u>13.437</u>	5.512	8.742
C-Mixup†	2.717	1.610	0.509	12.998	0.020	2.041	0.057	13.372	5.177	8.435
ADA†	2.360	1.373	0.515	13.128	0.021	2.116	0.059	13.464	-	-

Inspecting the data distribution and its modifications, reveals the differences between the data generated when select λ values are used. The distribution of the original data (green) shown in Fig. 3 (right) is bimodal. In comparison, the blue curve ($\lambda = 0$) for which the small singular values change to zeros, has a unimodal distribution. The orange curve ($\lambda = 0.5$) for which both the non-augmented model and a model trained with FOMA attained the minimum loss, has a smoother transition between the major mode and the minor mode. From the analysis above, we conclude the following. First, the network prefers data whose distribution is simpler (orange) than the original distribution (blue), yet not too simple (green). Second, our regularization encourages this tendency by providing the model with such data, leading to improved MSE profiles. To the best of our knowledge, the above analysis is novel on deep regression models.

Notably, while it may argued that the behavior in Fig. 3 (left) is natural and intuitive as the model "simply" performs better on denoised signals, we argue differently. In particular, this plot somewhat contradicts our understanding of overfitting which occurs in high probability for tiny datasets such as NO2 (the entire dataset is composed of 500 entries) using multiple weights network such as the fully connected network we used with around $20k$ parameters. Specifically, since the data is highly likely to be overfit by the network, we expect the MSE value to be lowest for $\lambda = 1$, and MSE value equal or higher for any $\lambda < 1$. Thus, we advocate that the above analysis may reveal a characteristic feature of regression neural networks. Our analysis is reinforced further as other datasets and architectures follow a similar pattern (App. B). Importantly, we are unaware of a similar experiment in the literature of deep regression neural networks.

4. Analysis

Relation to additive noise. In what follows, we would like to answer the following question: Does applying FOMA is merely a variant of injecting additive noise? To this end, we analyze FOMA from a perturbation theory viewpoint. Specifically, we would like to understand how a random data perturbation affects the singular values of the data matrix $A \in \mathbb{R}^{q\times r}$, $q \geq r$. We denote by $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$ the singular values of A. The perturbed matrix and its singular values set are denoted by $\tilde{A} = A + E$ and $\{\tilde{\sigma}_j\}_{j=1}^r$, respectively. We write $\inf_2(A)$ and $|A|_2$ to denote the smallest and largest singular values of any matrix A. The following classical result provides an estimated bound for the perturbed singular values (Stewart, 1979; 1998).

Theorem 1. *Let P be the orthogonal projection onto the column space of A. Let $P_\perp = I - P$. Then*

$$\tilde{\sigma}_j^2 = (\sigma_j + \gamma_j)^2 + \eta_j^2 \,, \quad j = 1\ldots, r \,,$$

where $|\gamma_j| \leq |P\,E|_2$ and $\inf_2(P_\perp E) \leq \eta_j \leq |P_\perp E|_2$.

Following (Stewart, 1979), we make two observations with respect to Thm. 1. First, if $\sigma_j \gg |E|_2$ then it dominates the bound and we have $\tilde{\sigma}_j \cong \sigma_j + \gamma_j$. Second and more important to our setting, when σ_j is of order $|E|_2$, the term η_j will tend to dominate. Indeed, in these cases the term η_j *increases* the singular value σ_j. We conclude that random perturbations to A tend to increase its small singular values. In contrast, FOMA typically decreases the small singular values, while leaving the large σ_j unchanged. Thus, FOMA is in effect a complementary approach to injecting additive noise, allowing a finer control over the resulting new samples. Finally, we note that for a certain choice of hyper-parameters, our approach can be viewed as injecting noise per the above analysis. For example, taking $\lambda \sim \text{Uniform}(1.0, \alpha)$ for $\alpha > 1.0$ will increase all the small singular values of A by a factor of $\lambda \in [1.0, \alpha]$, where Uniform is the random uniform distribution.

FOMA as a Vicinal Risk Minimization (VRM). Given a cost function $c : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$, the learning problem aims at minimizing the expectation of the loss $c(f(x), y)$ over the distribution $\mathcal{P}(x, y), x \in \mathcal{X}, y \in \mathcal{Y}$. A fundamental challenge, shared by most real-world scenarios, is that the true distribution of the data is unfortunately *unknown*. The alternative is to minimize over the empirical distribution of

a train set $\{(x_i, y_i)\}_{i=1}^n$ given by

$$\mathrm{d}\,\mathcal{P}_{\text{emp}}(x, y) = \frac{1}{n}\sum_i \delta_{x_i}(x)\delta_{y_i}(y)\ .$$

The resulting scheme is the common training procedure of modern neural networks, formally known as the Empirical Risk Minimization (ERM) (Vapnik, 1991).

While $\mathcal{P}_{\text{emp}}$ provides a basic approximation of the true $\mathcal{P}$, it was suggested (Chapelle et al., 2001) that other density estimates $\mathrm{d}\,\mathcal{P}_{\text{est}}$ that take into account the *vicinity* of (x_i, y_i) should be considered. The recent `mixup` approach (Guo et al., 2019) exploits this idea by proposing a Vicinal Risk Minimization (VRM) procedure that is based on the vicinal distribution estimate $\frac{1}{n}\sum_{i,j}\delta_{\tilde{x}_{ij}(\lambda)}(x)\delta_{\tilde{y}_{ij}(\lambda)}(y)$, defined using convex combinations $\tilde{z}_{ij}(\lambda) = \lambda z_i + (1-\lambda)z_j$ for $z \in \{x, y\}$ and $\lambda \sim \text{Beta}(\alpha, \alpha)$. In this context, the main difference between `FOMA` and `mixup` is in the definition of vicinity as we describe below.

We denote by $\mathcal{T}(x, y)$ the tangent plane of the data manifold $\mathcal{M}$ at the point $(x, y) \in \mathcal{M} \subset \mathcal{X} \times \mathcal{Y}$. Namely, $\mathcal{T}(x, y)$ is the linear approximation of $\mathcal{M}$ at (x, y). For every pair (x, y), we define a new density distribution $\mathcal{P}_{\text{tan}}$ which considers all pairs (a, b) in the tangent plane of $(u, v) \in \mathcal{M}$. Formally,

$$\mathrm{d}\,\mathcal{P}_{\text{tan}}(x, y) = \int_{\mathcal{M}}\int_{\mathcal{T}(u,v)} \delta_a(x)\delta_b(y)\,\mathrm{d}\,ab\,\mathrm{d}\,uv\ .$$

Then, `FOMA` approximates the latter expression by generating an estimate of the tangent plane $\mathcal{T}_{\text{est}}$ via `SVD`, yielding the following vicinal estimate

$$\mathrm{d}\,\mathcal{P}_{\text{est}}(x, y) = \frac{1}{n}\sum_i \frac{1}{k_i}\sum_j \delta_{x_j}(x)\delta_{y_j}(y)\ ,$$
$$(x_j, y_j) \in \mathcal{T}_{\text{est}}(x_i, y_i)\ ,\ k_i = |\mathcal{T}_{\text{est}}(x_i, y_i)|\ .$$

5. Experiments

5.1. In-Distribution Generalization

In this section, we assess the effectiveness of `FOMA` and compare it to previous methods in terms of its ability to generalize within the given distribution. We utilize the datasets used in the study conducted by Yao et al. (2022) and closely replicate their experimental setup.

Datasets. We use the following five datasets to evaluate the performance of in-distribution generalization. Two tabular datasets: Airfoil Self-Noise (Airfoil) (Brooks et al., 2014) and NO2 (Aldrin, 2004). Airfoil includes the aerodynamic and acoustic test results of airfoil blade sections and NO2 predicts the level of air pollution at specific locations. Two time series datasets: Exchange-Rate and Electricity (Lai et al., 2018), where Exchange-Rate provides a collection of daily exchange rates and Electricity is utilized for predicting the hourly electricity consumption. Finally, Echocardiogram Videos is designed for predicting the ejection fraction. It consists of a collection of videos that provide visual representations of the heart from various perspectives. See App. C for a detailed description of the datasets.

Experimental settings. We conduct a comparative analysis between our approach, `FOMA`, and several existing strong baseline methods, namely Mixup (Zhang et al., 2017), Manifold-Mixup (Verma et al., 2019), C-Mixup (Yao et al., 2022), Anchor Data Augmentation (ADA) (Schneider et al., 2023), and classical expected risk minimization (ERM). Importantly, we denote by C-Mixup* and ADA* the results for these methods as reproduced in our environment, whereas C-Mixup† and ADA† represent the results as reported in (Yao et al., 2022; Schneider et al., 2023), respectively. For a fair comparison, we compare our results with respect to the starred models, as we were unable to reproduce the reported results in the original papers; this observation also appeared in (Schneider et al., 2023) regarding C-Mixup. Further, to maintain consistency with the methodology outlined in (Yao et al., 2022), we adopt the same model architectures, using a fully connected three-layer network for tabular datasets, an LST-Attn (Lai et al., 2018) for time series data, and EchoNet-Dynamic (Ouyang et al., 2020) for predicting the ejection fraction. We use Root Mean Square Error (RMSE) and Mean Averaged Percentage Error (MAPE) as evaluation metrics. Detailed experimental settings and hyperparameters are provided in App. E.

Results. We report the in-distribution generalization results in Table 3. In all settings, lower numbers are better. Per column, we mark in bold the best available result, and we underline the second best error measure. Table 3 shows that both variants of `FOMA` improve over standard training via ERM, often by large margins. Further, our approach achieves new state-of-the-art error measures in comparison to the other baseline approaches on all datasets and metrics, except for Electricity where we obtain on-par results to C-Mixup. In particular, $\texttt{FOMA}_\rho$ yields the best available results in most cases. Finally, we observe the most significant improvement occurs on two datasets: Airfoil and Exchange-Rate in which `FOMA` reduces the error of the state-of-the-art result by approximately 37% and 38%, respectively.

5.2. Out-of-Distribution Robustness

In this section, we assess the effectiveness of `FOMA` and compare it to previous methods on tasks involving out-of-distribution robustness. To this end, we consider the datasets used in the study (Yao et al., 2022), and we closely replicate their experimental setup.

Table 4. Comparison of out-of-distribution robustness problems. Bold values represent the best results and underlined values are second best. We report the average RMSE across domains and the "worst within-domain" RMSE over three different seeds. For the DTI and PovertyMap datasets, we report average R and "worst within-domain" R. Full results with standard deviation can be found in App F.

	RCF (RMSE)	Crimes (RMSE)		DTI (R)		PovertyMap (R)	
	Avg. ↓	Avg. ↓	Worst ↓	Avg. ↑	Worst ↑	Avg. ↑	Worst ↑
ERM†	0.164	0.136	0.170	0.483	0.439	0.80	0.50
Mixup†	0.159	0.134	0.168	0.459	0.424	0.81	0.46
ManiMixup†	0.157	**0.128**	**0.155**	0.474	0.431	-	-
C-Mixup⋆	**0.153**	0.131	0.166	0.474	0.441	0.803	**0.516**
ADA⋆	0.171	1.30	0.156	-	-	-	-
FOMA	0.171	0.132	0.164	0.496	0.430	0.776	0.492
FOMA$_\rho$	0.159	**0.128**	0.158	**0.503**	**0.459**	**0.832**	0.482
C-Mixup†	0.146	0.123	0.146	0.498	0.458	0.81	0.53
ADA†	0.175	0.130	0.156	0.493	0.448	0.794	0.522

Table 5. Ablation study of FOMA over modifying data at the input or latent levels, different loss scaling profiles $\mu(\lambda)$, and scaling down the small or large singular values.

			NO2		Electricity	
mode	$\mu(\lambda)$	scale	RMSE↓	MAPE↓	RMSE↓	MAPE↓
input	1	small	$\mathbf{0.52 \pm 0.01}$	13.23 ± 0.09	$5.83e^{-2} \pm 1e^{-3}$	$\mathbf{13.02 \pm 0.07}$
input	λ	small	0.53 ± 0.01	13.23 ± 0.09	$5.83e^{-2} \pm 1e^{-3}$	13.02 ± 0.37
input	λ^2	small	0.54 ± 0.01	13.53 ± 0.27	$\mathbf{5.79e^{-2} \pm 2e^{-4}}$	13.50 ± 0.04
input	1	large	0.79 ± 0.02	19.22 ± 0.45	$5.89e^{-2} \pm 4e^{-4}$	13.89 ± 0.39
input	λ	large	0.76 ± 0.02	18.61 ± 0.37	$5.86e^{-2} \pm 9e^{-4}$	13.15 ± 0.05
input	λ^2	large	0.74 ± 0.01	18.01 ± 0.28	$5.88e^{-2} \pm 1e^{-3}$	13.23 ± 0.13
latent	1	small	0.53 ± 0.01	13.21 ± 0.08	$6.11e^{-2} \pm 9e^{-4}$	15.58 ± 0.38
latent	1	large	0.65 ± 0.02	16.35 ± 0.56	$7.16e^{-2} \pm 1e^{-3}$	18.62 ± 0.70
input + latent	1	small	0.52 ± 0.01	$\mathbf{13.19 \pm 0.14}$	$5.94e^{-2} \pm 1e^{-3}$	14.78 ± 0.48
input + latent	1	large	0.65 ± 0.02	21.25 ± 0.48	$7.97e^{-2} \pm 1e^{-4}$	20.77 ± 0.41

Datasets. We use the following four datasets to assess the performance of out-of-distribution robustness. **1)** RCFashionMNIST (RCF) (Yao et al., 2022), which is a synthetic modification of Fashion-MNIST, modeling sub-population shifts, where the goal is to predict the angle of rotation for each object. **2)** Communities and Crime (Crime) (Redmond, 2009) is a tabular dataset that focuses on predicting the total number of violent crimes per 100K population, where the objective is to develop a model that can generalize to states that were not included in the training data. **3)** Drug-Target Interactions (DTI) (Huang et al., 2021) is aiming to predict out-of-distribution drug-target interactions where the year is the domain information. **4)** PovertyMap (Koh et al., 2021) is a satellite image regression dataset that has been created with the goal of estimating asset wealth in countries that were not included in the training set. For further details on the various datasets, see App. C.

Experimental settings. Similarly to Sec. 5.1, we compare FOMA to the same baseline approaches. In terms of metrics, we report the RMSE (lower is better) for RCF-MNIST and Crimes. For PovertyMap and DTI, we use R (higher is better) as the evaluation metric, originally proposed in the respective papers (Koh et al., 2021; Huang et al., 2021). Following Yao et al. (2022), we trained ResNet-18 for RCF-MNIST and PovertyMap datasets, three-layer fully connected networks for Crimes, and DeepDTA (Öztürk et al., 2018) for DTI. More details regarding hyper-parameters and the experimental setup appear in App. E.

Results. We report both the average and worst-domain performance metrics for out-of-distribution tasks in Table 4. Dashed cells represent cases where results are not available. While the results for our approach are somewhat more mixed in comparison to the in-distribution challenge, we still find FOMA to be highly effective. In particular, FOMA improved over ERM in almost all cases, with the exception of RCF.

We find our technique to obtain comparable results for RCF and Crimes with respect to the best baselines. Notably, FOMA presents a noticeable gap in DTI and PovertyMap in comparison to SOTA approaches such as C-Mixup and ADA. Finally, similarly to the in-distribution setting, FOMA_ρ achieves better results in comparison to FOMA.

5.3. Ablation study

FOMA is a data augmentation method that scales down singular values of the data. However, there are several design choices to make. For example, we can choose to apply FOMA on the input data, on the learned representations, or apply it on both in succession. Another decision is what singular values we should scale down: the smaller ones, capturing less explained variance of the data, or the larger ones, which may create samples that are further away from the data manifold and thus expand the underlying distribution. Another choice to make is how to scale the singular values and potentially the loss function. The values $\lambda \sim \text{Beta}(\alpha, \alpha)$ by which we scale the singular values vary between batches, the higher the value of λ, the more noise is removed. By scaling the loss function differently for each batch, we can change the update rule of the optimizer, potentially leading to an improved behavior of FOMA.

We detail in Table 5 the ablation results we obtained for NO2 and Electricity datasets while exploring the parameter spaces of the above design choices. Overall, applying our technique at the input level, without scaling the loss, and scaling the small singular values seems to consistently yield good results across datasets and tasks (see also Tables 8, 9). More specifically, we find in Table 5 that for the datasets NO2 and Electricity, scaling down the small singular values is preferred to scaling the larger ones and more generally, this observation holds for all datasets (see Table 8). On the other hand, when using FOMA_ρ, allowing more freedom in selecting what singular values to scale, some datasets achieve better results when scaling the *larger* singular values, e.g., Airfoil and Exchange-rate. Furthermore, we have two observations regarding the loss scaling profiles $\mu(\lambda)$: 1) the effect of decreasing μ is inversely proportional between scaling small singular values and large singular values. For instance, when scaling the small singular values of NO2, smaller μ corresponds with better performance, whereas when scaling the large singular values, larger μ corresponds to better performance. 2) The effect of μ is inversely proportional between the datasets, decreasing μ improves the performance on Electricity while achieving worse performance on NO2. Finally, we note that employing FOMA in the latent space or both in the input and latent spaces yields inconsistent results across datasets.

6. Discussion

We have proposed FOMA, a data-driven method for data augmentation of regression tasks. We showed that FOMA supports the network tendency of representing dominant components of its input signals by creating virtual examples sampled from the tangent planes of the original train set. Implementing FOMA is straightforward, and it is fully differentiable. Throughout an extensive evaluation, we have shown that FOMA improves the generalization error of neural models on regression benchmarks including in-distribution generalization and out-of-distribution tasks. We also ablated our model with respect to several design choices. Generally, we find our method to obtain highly competitive results and often surpass state-of-the-art approaches.

We inspected the effect of the hyper-parameters α, ρ and whether to scale the small or large singular values. For FOMA, we observe a relatively consistent performance across tasks, whereas FOMA_ρ presents mixed results (see Tabs. 8, 9). Given that FOMA and FOMA_ρ perform similarly, we can not derive a specific guideline for choosing these hyper-parameters. Another limitation is that the time complexity of FOMA is governed by the SVD calculation, which may be restrictive for large train batches. Finally, we mention that if $k = n_0 + m$, i.e., the tangent space dimension equals that of the data rank, then our approach can be only applied to the dominant singular values.

There are several exciting avenues for future exploration. First, is there a fundamental link between the vicinal distribution employed and the learned representation? While several existing works suggest that *linearity* yields better models (Azencot et al., 2020; Berman et al., 2023; Zeng et al., 2023), the model dependency on the specific definition of vicinity is still not well understood. Second, can similar methods to ours be useful in classification tasks? The adaptation of FOMA to classification is straightforward, however, several design choices which were tuned for regression may require change in a classification setting. In particular, the computational demands of SVD-based data augmentation are higher in comparison to mixup schemes. Improving these aspects by e.g., approximating the tangent space of the manifold may be highly impactful in regression as well as classification tasks. Another interesting avenue to explore is the relation between FOMA and generative modeling (Naiman et al., 2024). We plan to explore these questions in future work.

Acknowledgements

This research was partially supported by the Lynn and William Frankel Center of the Computer Science Department, Ben-Gurion University of the Negev, an ISF grant 668/21, an ISF equipment grant, and by the Israeli Council

for Higher Education (CHE) via the Data Science Research Center, Ben-Gurion University of the Negev, Israel.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Aldrin, M. Cmu statlib dataset. http://lib.stat.cmu.edu/datasets/, 2004. DOI: https://doi.org/10.24432/C5VW2C.

Allen-Zhu, Z., Li, Y., and Liang, Y. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32, 2019a.

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019b.

Azencot, O., Erichson, N. B., Lin, V., and Mahoney, M. Forecasting sequential data using consistent koopman autoencoders. In *International Conference on Machine Learning*, pp. 475–485. PMLR, 2020.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Berman, N., Naiman, I., and Azencot, O. Multifactor sequential disentanglement via structured koopman autoencoders. In *The Eleventh International Conference on Learning Representations, ICLR*, 2023.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Brooks, T., Pope, D., and Marcolini, M. Airfoil Self-Noise. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5VW2C.

Cayton, L. et al. *Algorithms for manifold learning*. eScholarship, University of California, 2008.

Chapelle, O., Weston, J., Bottou, L., and Vapnik, V. Vicinal risk minimization. *Advances in neural information processing systems*, pp. 416–422, 2001.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. AutoAugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. RandAugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

DeVries, T. and Taylor, G. W. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*, 2017.

Devroye, L., Györfi, L., and Lugosi, G. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

Dubost, F., Bortsova, G., Adams, H., Ikram, M. A., Niessen, W., Vernooij, M., and de Bruijne, M. Hydranet: data augmentation for regression neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 438–446. Springer, 2019.

Eckart, C. and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Facco, E., d'Errico, M., Rodriguez, A., and Laio, A. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific reports*, 7(1):12140, 2017.

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

Greenewald, K., Gu, A., Yurochkin, M., Solomon, J., and Chien, E. k-mixup regularization for deep learning via optimal transport. *arXiv preprint arXiv:2106.02933*, 2021.

Guo, H., Mao, Y., and Zhang, R. Mixup as locally linear out-of-manifold regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3714–3722, 2019.

Hanin, B. and Sun, Y. How data augmentation affects optimization for linear regression. *Advances in Neural Information Processing Systems*, 34, 2021.

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

Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y., Leskovec, J., Coley, C. W., Xiao, C., Sun, J., and Zitnik, M. Therapeutics data commons: machine learning datasets and tasks for therapeutics. *arXiv preprint arXiv:2102.09548*, 2021.

Hwang, S.-H. and Whang, S. E. RegMix: Data mixing augmentation for regression. *arXiv preprint arXiv:2106.03374*, 2021.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Ionescu, C., Vantzos, O., and Sminchisescu, C. Matrix backpropagation for deep networks with structured layers. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2965–2973, 2015.

Kaufman, I. and Azencot, O. Data representations' study of latent image manifolds. In *International Conference on Machine Learning*, pp. 15928–15945. PMLR, 2023.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lemley, J., Bazrafkan, S., and Corcoran, P. Smart augmentation learning an optimal data augmentation strategy. *Ieee Access*, 5:5858–5869, 2017.

Lim, S., Kim, I., Kim, T., Kim, C., and Kim, S. Fast AutoAugment. *Advances in Neural Information Processing Systems*, 32, 2019.

Lim, S. H., Erichson, N. B., Utrera, F., Xu, W., and Mahoney, M. W. Noisy feature mixup. *arXiv preprint arXiv:2110.02180*, 2021.

Ma, Y. and Fu, Y. *Manifold learning theory and applications*, volume 434. CRC press Boca Raton, FL, 2012.

Muthukumar, V., Narang, A., Subramanian, V., Belkin, M., Hsu, D., and Sahai, A. Classification vs regression in overparameterized regimes: Does the loss function matter? *Journal of Machine Learning Research*, 22(222): 1–69, 2021.

Naiman, I. and Azencot, O. An operator theoretic approach for analyzing sequence neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 9268–9276, 2023.

Naiman, I., Berman, N., and Azencot, O. Sample and predict your latent: modality-free sequential disentanglement via contrastive estimation. In *International Conference on Machine Learning*, pp. 25694–25717. PMLR, 2023.

Naiman, I., Erichson, N. B., Ren, P., Mahoney, M. W., and Azencot, O. Generative modeling of regular and irregular time series data via koopman VAEs. In *The Twelfth International Conference on Learning Representations, ICLR*, 2024.

Ouyang, D., He, B., Ghorbani, A., Yuan, N., Ebinger, J., Langlotz, C. P., Heidenreich, P. A., Harrington, R. A., Liang, D. H., Ashley, E. A., et al. Video-based ai for beat-to-beat assessment of cardiac function. *Nature*, 580 (7802):252–256, 2020.

Öztürk, H., Özgür, A., and Ozkirimli, E. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Redmond, M. Communities and Crime. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C5VW2C.

Rothenhäusler, D., Meinshausen, N., Bühlmann, P., and Peters, J. Anchor regression: Heterogeneous data meet causality. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(2):215–246, 2021.

Schneider, N., Goshtasbpour, S., and Perez-Cruz, F. Anchor data augmentation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Shorten, C. and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

Shorten, C., Khoshgoftaar, T. M., and Furht, B. Text data augmentation for deep learning. *Journal of big Data*, 8 (1):1–34, 2021.

Singer, A. and Wu, H.-T. Vector diffusion maps and the connection laplacian. *Communications on pure and applied mathematics*, 65(8):1067–1144, 2012.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Stewart, G. A note on the perturbation of singular values. *Linear Algebra and Its Applications*, 28:213–216, 1979.

Stewart, G. W. Perturbation theory for the singular value decomposition. Technical report, 1998.

Tian, K., Lin, C., Sun, M., Zhou, L., Yan, J., and Ouyang, W. Improving auto-augment via augmentation-wise weight sharing. *Advances in Neural Information Processing Systems*, 33:19088–19098, 2020.

Vapnik, V. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pp. 6438–6447. PMLR, 2019.

Wang, Y., Pan, X., Song, S., Zhang, H., Huang, G., and Wu, C. Implicit semantic data augmentation for deep networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., and Xu, H. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

Wu, S., Zhang, H., Valiant, G., and Ré, C. On the generalization effects of linear transformations in data augmentation. In *International Conference on Machine Learning*, pp. 10410–10420. PMLR, 2020.

Yao, H., Wang, Y., Zhang, L., Zou, J., and Finn, C. C-mixup: Improving generalization in regression. In *Proceeding of the Thirty-Sixth Conference on Neural Information Processing Systems*, 2022.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

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

A. Method overview

In this section we provide details for different methods used to evaluate the intrinsic dimension and linear dimension which are used for FOMA and FOMAρ respectively.

A.1. Intrinsic dimension

To estimate the intrinsic dimension of data, we use the TwoNN (Facco et al., 2017) ID estimator. The ID-estimator relies on the distances to only the two closest neighbors of each point, minimizing the influence of inconsistencies in the dataset during estimation.
Let $X = \{x_1, x_2, \cdots, x_N\}$ be a set of points sampled uniformly on a manifold with intrinsic dimension d. For each point x_i, we calculate the two shortest distances r_1, r_2 from other elements in $X \setminus \{x_i\}$ and determine the ratio $\mu_i = \frac{r_2}{r_1}$. It has been proven that $\mu_i, 1 \leq i \leq N$ are distributed according to a Pareto distribution with parameter $d+1$ on the interval $[1, \infty)$, specifically $f(\mu_i \mid d) = d\mu_i^{-(d+1)}$. While d can be estimated by maximizing the likelihood:

$$P(\mu_1, \mu_2, \cdots \mu_N \mid d) = d^N \prod_{i=1}^{N} \mu_i^{-(d+1)} \,. \tag{1}$$

We adopt the approach suggested by Facco et al. (2017) using the cumulative distribution $F(\mu) = 1 - \mu^{-d}$. The method involves estimating the parameter d through linear regression on the empirical estimate of $F(\mu)$. To do this, we arrange the values of μ in ascending order and define $F^{emp}(\mu_i) \approx \frac{i}{N}$. A linear regression is then performed on the set of points $\{(\log \mu_i, -\log(1 - F_i^{emp}))\}_{i=1}^{N}$ where the slope of the fitted line is the estimated ID.

A.2. Linear dimension

A common method for estimating the linear dimension of data is to perform principal component analysis (PCA) or SVD and count the number of components that should be included to describe some percentage of the variance in the data, usually above 90%. More formally:

$$k = \arg\max_{\tilde{k}} \sum_{j=1}^{\tilde{k}} \sigma_j / \sum_j \sigma_j \leq \rho \,.$$

B. Sequential models capture dominant components of data better

Following the discussion in Sec. 3, we verify empirically that neural networks model the dominant parts of their data better. We repeat the experiment in Fig. 3 (left) using additional two datasets which are trained on different architectures. For evaluation, we use the dataset whose singular values are modified using varying values of λ. The results are presented in Fig. 4, where solid lines represent the results of the non-regularized model, and dashed lines are associated with models trained with FOMA. Remarkably, we observe a similar qualitative behavior as we reported in Sec. 3. In particular, the highest MSE values are obtained for both the baseline and regularized models for $\lambda = 1$, i.e., when the data is unchanged. Further, the model attain improved error measures as λ decreases, where the error profile is similar for the baseline and regularized models. Based on these results, we deduce that sequential models prefer to represent and compute the dominant components of data, reinforcing our choice for supplying such data to the network during training.

C. Dataset Description

In this section, we provide detailed descriptions of datasets used in the experiments in this work.

Airfoil Self-Noise (Brooks et al., 2014). The dataset comprises aerodynamic and acoustic test findings for various sizes of NACA 0012 airfoils, obtained at different wind tunnel speeds and angles of attack. Each input instance consists of five features: frequency, angle of attack, chord length, free-stream velocity, and suction side displacement thickness. The label represents the one-dimensional scaled sound pressure level. To normalize the input features, min-max normalization is applied. As per reference (Hwang & Whang, 2021), the training, validation, and test sets consist of 1003, 300, and 200 examples, respectively.

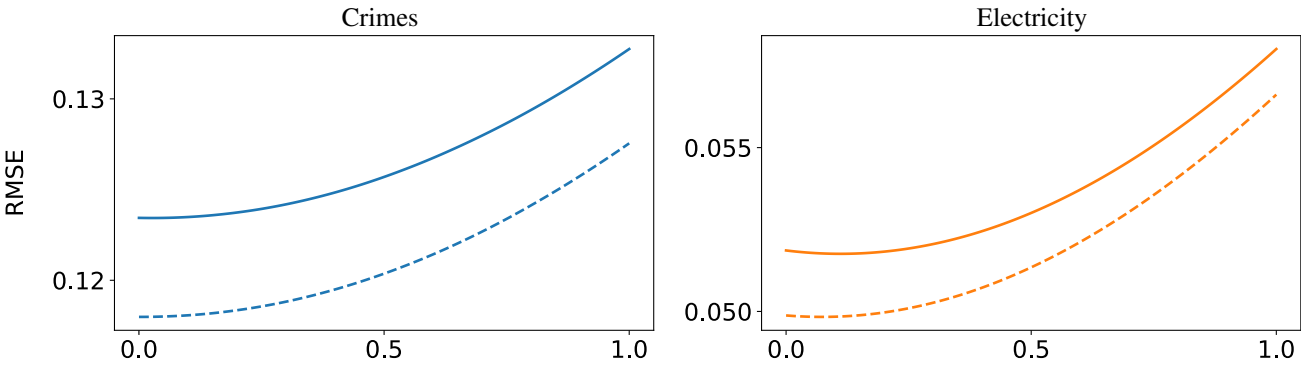

Figure 4. Evaluating a non-augmented model (solid lines) and a model trained with FOMA (dashed lines) on train data whose small singular values are scaled down for different values of λ (see also Fig. 3, left). Communities and crime dataset trained on a three-layer full connected network (left). Electricity trained on LST-Attn (Lai et al., 2018) (right).

NO2 (Aldrin, 2004). The NO2 emission dataset originates from a study examining the relationship between air pollution near a road and traffic volume along with meteorological variables. Each input comprises seven features, including the logarithm of the number of cars per hour, temperature 2 meters above ground, wind speed, temperature difference between 25 and 2 meters above ground, wind direction, hour of the day, and the day number since October 1st, 2001. The hourly values of the logarithm of NO2 concentration, measured at Alnabru in Oslo between October 2001 and August 2003, serve as the response variable or label. As per reference (Hwang & Whang, 2021), there are 200 examples in the training set, 200 in the validation set, and 100 in the test set.

Exchange-Rate (Lai et al., 2018). The exchange-rate dataset comprises a time-series collection of daily exchange rates for eight countries: Australia, Britain, Canada, Switzerland, China, Japan, New Zealand, and Singapore, spanning from 1990 to 2016. The total length of the time series is 7,588, with a daily sampling frequency. A sliding window size of 168 days is applied. The input dimension is 168×8, and the label dimension is 1×8 data points. Following the methodology outlined in (Lai et al., 2018), the dataset has been divided into training (60%), validation (20%), and test (20%) sets in chronological order.

Electricity (Brooks et al., 2014). This dataset is a time-series collection obtained from 321 clients, representing electricity consumption in kWh recorded every 15 minutes from 2012 to 2014. The total length of the time series is 26,304, sampled hourly. As with the Exchange-Rate data, a window size of 168 is utilized, resulting in an input dimension of 168×321 and a corresponding label dimension of 1×321. Similar to the methodology described in Lai et al. (Lai et al., 2018), the dataset is divided accordingly.

Echo (Ouyang et al., 2020). The Echocardiogram Videos dataset comprises 10,030 labeled apical-4-chamber echocardiogram videos captured from various perspectives, accompanied by expert annotations for studying cardiac motion and chamber sizes. These videos were sourced from individuals undergoing imaging at Stanford University Hospital between 2016 and 2018. To delineate the left ventricle area, initial preprocessing involves frame-by-frame semantic segmentation of the videos. This preprocessing method generates video clips containing 32 frames of 112×112 RGB images, which serve as input for predicting ejection fraction. The dataset is partitioned into training, validation, and test sets, with sizes of 7,460, 1,288, and 1,276, respectively.

RCF. RCF-MNIST, where "RCF" stands for "Rotated-Colored-Fashion", is a dataset constructed with specific color and rotation attributes. In this dataset, the normalized RGB vector for red and blue is $[1, 0, 0]$ and $[0, 0, 1]$ respectively, and the normalized rotation angle (i.e., label) for each image is denoted as g, where $g \in [0, 1]$. During the construction of the training set, 80% of the images are colored using the RGB value $[g, 0, 1 - g]$, while the remaining 20% are colored with $[1 - g, 0, g]$. Consequently, there is a strong spurious correlation between color information and labels within the training set. To simulate distribution shift in the test set, this spurious correlation is reversed, with 80% of the images colored using RGB values $[1 - g, 0, g]$, and the remaining 20% with $[g, 0, 1 - g]]$. The impact of this spurious correlation on performance is evaluated by comparing the performance of the same test set with or without distribution shift. The results, presented in

Table 11, demonstrate that the subpopulation shift induced by the spurious correlation indeed affects performance negatively, as anticipated.

PovertyMap. This dataset is part of the WILDS benchmark (Koh et al., 2021), comprising satellite images sourced from 23 African countries, which are utilized for predicting the village-level real-valued asset wealth index. Each input consists of a 224×224 multispectral LandSat satellite image with 8 channels, while the corresponding label represents the real-valued asset wealth index. The domains of the images encompass information regarding the country, urban, and rural areas. The dataset is divided into 5 distinct cross-validation folds, with all countries in these splits being disjoint to facilitate the out-of-distribution setting. All experimental configurations adhere to the methodology outlined by Koh et al. (Koh et al., 2021).

Crime (Redmond, 2009). The Communities And Crimes dataset is a tabular compilation that integrates socio-economic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR. It encompasses 122 attributes believed to have some plausible connection to crime, such as median family income and the percentage of officers assigned to drug units. The target attribute for prediction is per capita violent crimes, which includes offenses such as murder, rape, robbery, assault, among others. To prepare the data, all numeric features are normalized using the decimal range 0.00 to 1.00 through an equal-interval binning method, and missing values are imputed with the average values of the corresponding attributes. Domain information is denoted by state identifications, resulting in a total of 46 domains. The dataset is partitioned into training, validation, and test sets containing 1,390, 231, and 373 instances, respectively. These sets consist of 31, 6, and 9 disjoint domains, respectively.

DTI (Huang et al., 2021). The Drug-target Interactions dataset is designed to forecast the binding activity score between each small molecule and its corresponding target protein. Input features encompass information on both the drug and target protein, represented by one-hot vectors, while the output label denotes the binding activity score. The training and validation sets are curated from the years 2013 to 2018, while the test set spans the years 2019 to 2020. The ”Year” attribute serves as the domain information.

D. Additional Experiments

D.1. Comparison with additive noise

In Section. 4 we show a relation of our method to additive noise. For completeness, we add a supplementary experiment that compares our method with additive noise as shown in Table. 6 and Table. 7. The experiment was conducted as follows: Given a sample $x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}^m$ we sample $\epsilon_x \sim \mathcal{N}(0, \sigma \times I_n)$, $\epsilon_y \sim \mathcal{N}(0, \sigma \times I_m)$ and create the noised sample $(\tilde{x_i} = x_i + \epsilon_x, y_i = \tilde{y_i} + \epsilon_y)$. Where σ is selected from $\{0.1, 0.01, 0.001, 0.0001\}$ and the best model is chosen according to the performance on the validation set.

Table 6. Comparison of in-distribution generalization tasks with additive noise.

	Airfoil		NO2		Exchange-Rate		Electricity	
	RMSE ↓	MAPE (%)↓	RMSE ↓	MAPE (%)↓	RMSE ↓	MAPE (%)↓	RMSE ↓	MAPE (%)↓
ERM†	2.901	1.753	0.537	13.615	0.0236	2.423	0.0581	13.861
ERM+noise	3.172	1.864	0.524	13.326	0.016	1.568	0.600	14.068
FOMA	1.646	0.963	0.521	13.23	0.013	1.280	0.058	14.614
FOMA$_\rho$	**1.471**	**0.816**	**0.512**	**12.894**	**0.013**	**1.262**	0.059	13.437

Table 7. Comparison of out-of-distribution robustness problems with additive noise.

	RCF (RMSE)	Crimes (RMSE)		DTI (R)	
	Avg. ↓	Avg. ↓	Worst ↓	Avg. ↑	Worst ↑
ERM†	0.164	0.136	0.170	0.483	0.439
ERM + noise	0.180	0.136	0.166	0.492	0.442
FOMA	0.171	0.132	0.164	0.496	0.430
FOMA$_\rho$	0.159	**0.128**	0.158	**0.503**	**0.459**

E. Hyperparameters

We list the hyperparameters for every dataset in Table 8 and Table 9 for the methods `FOMA` and `FOMA`ρ, respectively. In our main results, we apply our method on the input space or on the latent space or both and report the one with best performance. All hyperparameters are selected by cross-validation, evaluated on the validation set. Some of the hyperparameters such as architecture and optimizer are not included in the table since they were not changed and were used as they appear in previous works (Yao et al., 2022).

Table 8. Hyperparameter choices for the experiments using `FOMA`.

Dataset	Airfoil	NO2	Exchange-Rate	Electricity	Echo	RCF	Crimes	PovertyMap	DTI
Learning rate	$5e^{-4}$	$1e^{-3}$	$1e^{-2}$	$5e^{-4}$	$1e^{-4}$	$1e^{-4}$	$5e^{-3}$	$1e^{-3}$	$5e^{-4}$
Batch size	32	64	16	128	10	128	8	32	64
Input/Latent	latent	both	input	both	latent	latent	input	latent	latent
Epochs	200	100	200	150	20	250	200	50	200
Singular values	small	small	small	small	small	small	small	small	large
α	0.8	0.9	0.2	1.1	1.1	1	0.6	1.5	0.5

Table 9. Hyperparameter choices for the experiments using FOMAρ.

Dataset	Airfoil	NO2	Exchange-Rate	Electricity	Echo	RCF	Crimes	PovertyMap	DTI
Learning rate	$5e^{-4}$	$1e^{-3}$	$1e^{-3}$	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$	$1e^{-3}$	$5e^{-3}$	$1e^{-2}$
Batch size	128	8	8	8	10	8	64	32	64
Input/Latent	input	input	input	input	latent	latent	both	latent	latent
Epochs	100	250	150	200	20	250	100	50	250
Singular values	large	small	large	large	small	small	large	small	large
α	1.4	0.3	1	0.7	1.1	1.5	0.6	1	0.6
ρ	0.975	0.95	0.8	0.875	0.85	0.95	0.875	0.875	0.825

F. Results with Standard Deviation

In Tables 10, 11, we report the full results of in-distribution generalization and out-of-distribution robustness respectively.

Table 10. Full results of in-distribution generalization. We compute the mean and standard deviation for results of three seeds.

	Airfoil		NO2		Exchange-Rate	
	RMSE ↓	MAPE (%)↓	RMSE ↓	MAPE (%)↓	RMSE ↓	MAPE (%)↓
ERM	$\underline{2.901 \pm 0.067}$	1.753 ± 0.078	0.537 ± 0.005	13.615 ± 0.165	0.023 ± 0.003	2.423 ± 0.365
Mixup	3.730 ± 0.190	2.327 ± 0.159	0.528 ± 0.005	13.534 ± 0.125	0.023 ± 0.002	2.441 ± 0.286
Mani Mixup	3.063 ± 0.113	1.842 ± 0.114	0.522 ± 0.008	13.357 ± 0.214	0.024 ± 0.004	2.475 ± 0.346
C-Mixup*	2.739 ± 0.06	1.640 ± 0.069	0.516 ± 0.01	13.069 ± 0.294	0.024 ± 0.005	2.455 ± 0.629
ADA*	2.357 ± 0.118	1.377 ± 0.064	$\underline{0.515 \pm 0.006}$	13.128 ± 0.12	0.021 ± 0.006	2.250 ± 0.781
FOMA	$\underline{1.646 \pm 0.103}$	$\underline{0.963 \pm 0.056}$	0.521 ± 0.013	13.23 ± 0.289	$\underline{0.013 \pm 0.000}$	$\underline{1.280 \pm 0.037}$
FOMA$_\rho$	$\mathbf{1.471 \pm 0.047}$	$\mathbf{0.816 \pm 0.008}$	$\mathbf{0.512 \pm 0.008}$	$\mathbf{12.894 \pm 0.217}$	$\mathbf{0.013 \pm 0.000}$	$\mathbf{1.262 \pm 0.037}$
C-Mixup†	2.717 ± 0.067	1.610 ± 0.085	0.509 ± 0.006	12.998 ± 0.271	0.0203 ± 0.001	2.041 ± 0.134
ADA†	2.360 ± 0.133	1.373 ± 0.056	0.514 ± 0.007	13.127 ± 0.146	0.020 ± 0.006	2.115 ± 0.689

	Electricity		Echo	
	RMSE ↓	MAPE (%)↓	RMSE ↓	MAPE (%)↓
ERM	0.058 ± 0.001	13.861 ± 0.152	5.402 ± 0.024	8.700 ± 0.015
Mixup	0.058 ± 0.000	14.306 ± 0.048	5.393 ± 0.040	8.838 ± 0.108
Mani Mixup	0.058 ± 0.000	14.556 ± 0.057	5.482 ± 0.066	8.955 ± 0.082
C-Mixup*	$\mathbf{0.057 \pm 0.000}$	13.471 ± 0.15	5.483 ± 0.097	9.121 ± 0.208
ADA*	0.059 ± 0.001	13.578 ± 0.146	-	-
FOMA	$\underline{0.058 \pm 0.000}$	$\underline{14.653 \pm 0.166}$	$\mathbf{5.215 \pm 0.061}$	8.331 ± 0.088
FOMA$_\rho$	0.059 ± 0.000	$\mathbf{13.437 \pm 0.26}$	5.476 ± 0.01	$\mathbf{8.742 \pm 0.091}$
C-Mixup†	0.057 ± 0.001	13.372 ± 0.106	5.177 ± 0.036	8.435 ± 0.089
ADA†	0.058 ± 0.001	13.464 ± 0.296	-	-

Table 11. Full results of out-of-distribution generalization. We compute the mean and standard deviation for results of three seeds.

	RCF (RMSE)	Crimes (RMSE)		DTI (R)		PovertyMap (R)	
	Avg. ↓	Avg. ↓	Worst ↓	Avg. ↑	Worst ↑	Avg. ↑	Worst ↑
ERM	0.162 ± 0.003	0.134 ± 0.003	0.173 ± 0.009	0.464 ± 0.014	0.429 ± 0.004	0.80 ± 0.04	$\underline{0.50 \pm 0.07}$
Mixup	0.176 ± 0.003	0.128 ± 0.002	$\mathbf{0.154 \pm 0.001}$	0.465 ± 0.004	0.437 ± 0.016	$\underline{0.81 \pm 0.04}$	0.46 ± 0.03
ManiMixup	0.157 ± 0.020	0.128 ± 0.003	$\underline{0.155 \pm 0.009}$	0.474 ± 0.004	0.431 ± 0.009	-	-
C-Mixup*	$\mathbf{0.153 \pm 0.004}$	$\underline{0.130 \pm 0.003}$	0.161 ± 0.01	0.475 ± 0.013	0.440 ± 0.016	0.804 ± 0.03	$\mathbf{0.517 \pm 0.06}$
ADA*	0.171 ± 0.009	0.130 ± 0.003	0.156 ± 0.006	-	-	-	-
FOMA	0.171 ± 0.015	0.132 ± 0.002	0.164 ± 0.002	$\underline{0.492 \pm 0.003}$	$\underline{0.442 \pm 0.019}$	0.776 ± 0.03	0.492 ± 0.05
FOMA$_\rho$	$\underline{0.159 \pm 0.01}$	$\mathbf{0.128 \pm 0.004}$	0.158 ± 0.002	$\mathbf{0.503 \pm 0.008}$	$\mathbf{0.459 \pm 0.01}$	$\mathbf{0.832 \pm 0.04}$	0.482 ± 0.06
C-Mixup†	0.146 ± 0.005	0.123 ± 0.000	0.146 ± 0.002	0.498 ± 0.008	0.458 ± 0.004	0.81 ± 0.03	0.53 ± 0.07
ADA†	0.162 ± 0.014	0.129 ± 0.003	0.155 ± 0.006	0.492 ± 0.009	0.448 ± 0.009	0.793 ± 0.03	0.521 ± 0.06