KERDEQ: OPTIMIZATION INDUCED DEEP EQUILIBRIUM MODELS VIA GAUSSIAN KERNEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Although the optimization induced deep equilibrium models (OptEqs) show the connection between the neural networks' structure and the designed hidden optimization problems (problems that the network's forward procedure tries to solve), we find that the linear kernels used in their hidden optimization problem hinder their performance, since linear kernels cannot extract non-linear feature dependencies from the inputs. Inspired by the classical machine learning algorithms, we use the widely used Gaussian kernels to construct the OptEqs hidden optimization problem and then propose our deep equilibrium model named KerDEQ. With Gaussian kernels, it can extract the input features' non-linear information more efficiently compared with the original OptEqs. Furthermore, KerDEQ can be regarded as a weight-tied neural network with infinite width and depth, therefore it shows better performance. Apart from that, our KerDEQ also shows better uncertainty calibration properties and performs more stably under different corruptions, especially under noise credit to the Gaussian kernel hidden optimization problem and its induced structure. Furthermore, we also conduct various experiments to demonstrate the effectiveness and reliability of our KerDEQ.

1 INTRODUCTION

Deep Neural Networks (DNNs) show impressive performance in many real-world tasks on various data like graphs (Scarselli et al., 2008), images (Redmon et al., 2016; He et al., 2016), sequences (Chowdhary, 2020), and others. However, most neural networks structure are constructed by experience or searching on the surrogate datasets (Liu et al., 2018; Zoph & Le, 2016). Therefore, these architectures can hardly be interpretable and such a phenomenon hinders further development. Apart from the current neural network models, traditional machine learning methods like Dictionary Learning (Tošić & Frossard, 2011; Mairal et al., 2009), Subspace Clustering (You et al., 2016) and other methods (Liu et al., 2014; Zhang et al., 2018; 2015; Liu & Li, 2016) can design their whole procedure by designing optimization problems with specific regularizers customized from their mathematical modeling and requirements. Thus, these models are easily interpreted. However, the traditional machine learning algorithms' whole procedures do not consider the hidden properties of features and labels. Therefore, they usually perform worse on tasks with more data.

To link two types of models, the optimization induced deep equilibrium models (OptEqs) (Xie et al., 2021) tries to recover the model's hidden optimization problem to make their model "mathematically explainable". As the formulation shows, the output features $\tilde{\mathbf{z}}^*$ for input $\boldsymbol{x}$ before the final prediction is obtained by solving the fixed point equation (Eqn (1)'s left part). Since the fixed point equation is also the first order stationary condition for problem Eqn (2), $\tilde{\mathbf{z}}^*$ is also the optimal solution for the following optimization problems.

$$\tilde{\mathbf{z}}^* = \tilde{\mathbf{W}}^\top \sigma(\tilde{\mathbf{W}}\tilde{\mathbf{z}}^* + \mathbf{U}\mathbf{x} + \mathbf{b}),\ \mathbf{y} = \tilde{\mathbf{W}}_c\tilde{\mathbf{z}}^* + \mathbf{b}_c, \tag{1}$$

$$\min_{\tilde{\mathbf{z}}} G(\tilde{\mathbf{z}}; \mathbf{x}) = \min_{\tilde{\mathbf{z}}} \left[\mathbf{1}^\top \ f(\tilde{\mathbf{W}}^{-1^\top}\tilde{\mathbf{z}}) - \left\langle \mathbf{U}\mathbf{x} + \mathbf{b}, \tilde{\mathbf{W}}^{-1^\top}\tilde{\mathbf{z}} \right\rangle + \frac{1}{2}\|\tilde{\mathbf{W}}^{-1^\top}\tilde{\mathbf{z}}\|^2 - \frac{1}{2}\|\tilde{\mathbf{z}}\|^2\right], \tag{2}$$

where $\mathbf{U}, \tilde{\mathbf{W}}, \mathbf{b}, \tilde{\mathbf{W}}_c, \mathbf{b}_c$ are learnable parameters. After solving the fixed point equation in the front of Eqn (1) for input $\mathbf{x}$ to get the output feature $\mathbf{z}^* \in \mathbb{R}^d$, we can get the prediction $\mathbf{y} \in \mathbb{R}^d$ with a linear projection. They call the fixed point equation as the OptEqs' equilibrium equation, which is the most important part of the equilibrium models and $\mathbf{z}^*$ is the equilibrium state. Since the forward

propagation of OptEqs as trying to solve the optimization problem defined above, researchers can also explain the structure of OptEqs by understanding the optimization problem or constructing new models by designing different problems for various tasks.

In order to improve the performance for OptEqs, the Multi-branch optimization induced deep equilibrium models (MOptEqs) (Li et al., 2021) construct a new deep equilibrium model by encoding the multi-scale inductive bias into the original OptEqs model. However, their performance is still not so good on image classification tasks although they enjoy satisfying "mathematical interpretability".

We notice that such models' hidden optimization problem can be divided into two parts: the output features' regularizer term and the feature extraction term. The latter is the most important since it is input-dependent and can influence the pattern extracted from input features. However, the former works only use the simplest linear kernel to extract features from the original inputs. Since the linear kernel can only work well when dealing with linear features, the representation power for original OptEqs is not enough. Motivated by the above statements, in our work, we take a further step by using the widely used Gaussian kernel to extract features from the inputs, which can effectively extract non-linear information compared with linear kernels and achieves better results on different kinds of tasks.

With Gaussian kernels, we propose our new optimization induced model, the optimization induced equilibrium model with Gaussian kernels (KerDEQ). It can also be regarded as OptEqs with weight-tied "infinite wide" projections because they are parameterized only by the models' weight parameter. Since OptEqs can be regarded as a weight-tied "infinite deep" model, our model can be regarded as a "double-infinite" model with better representation abilities. Therefore, our KerDEQ enjoys better representative abilities. Apart from the above findings, we also find that our KerDEQ's similarity between two samples is bounded by a stationary function which is mainly rely on the samples distance. Thereby, KerDEQ can hardly be overconfident on unseen samples and enjoy better uncertainty calibration ability in our analysis and experiments. In other words, our KerDEQ can hardly make over-confident predictions on unseen samples and can return more confident predictions on learned samples. For this reason, our KerDEQ is more reliable compared with the original OptEqs. We summarize our contributions as follows:

- We first reformulate the OptEqs' hidden optimization problem with Gaussian kernels. Then we propose our new DEQ model KerDEQ. Compared with the former OptEqs, our KerDEQ can still enjoy mathematical interpretability with better performance on real-world datasets.
- We have analyzed the theoretical properties of our KerDEQ and find that our KerDEQ can be regarded as a weight-tied neural network with both infinite width and depth. Empirical results also confirm the superiority of our KerDEQ.
- Through the similarity analysis for our KerDEQs' output for two samples, we can demonstrate that our KerDEQ can enjoy better uncertainty calibration performance against traditional OptEqs. We also conducted the out-of-distribution (OOD) uncertainty calibration experiments. And the empirical results also confirm our results.

2 Related Works

2.1 Implicit Models

Most modern deep learning approaches provide explicit computation graphs for forward propagations and we call these models "*explicit* models". Contrary to these models, recent researchers proposed some neural architecture with dynamic computation graphs and we call them "*implicit* models". For example, Neural ODEs Chen et al. (2018b) encode their neural architectures by a differential system with learnable parameters, then the implicit ODE solvers they used are equivalent to a continuous ResNet taking infinitesimal steps. Since the whole structure can be interpreted by differential systems, it leverages the black box of traditional neural networks. Because of the flexibility and the interpretability of implicit models, the design of implicit models draws much attention these days ((Ghaoui et al., 2019; Gould et al., 2019)). Many kinds of implicit models have been proposed, including optimization layers (Djolonga & Krause, 2017; Amos & Kolter, 2017), differentiable physics engines (Qiao et al., 2020; de Avila Belbute-Peres et al., 2018), logical structure

learning (Wang et al., 2019),differential programming (Xie et al., 2019; Scarselli et al., 2008; Chen et al., 2022; Bai et al., 2020; 2019).

Among the implicit models, OptEqs proposed by Xie et al. (2021) not only achieve superior performance against other implicit models but also explicitly explain the relationships between its structure and a well-defined optimization problem. Its forward propagation is to acquire the fixed points for Eqn (1), which can also be regarded as an "infinite" layer model. Other DEQ models with multi branches (Li et al., 2021; Bai et al., 2020) are designed for multi-scale image inputs and show better performance than other DEQ. However, their performance is not satisfying on some large datasets.

2.2 Infinite Wide Models and Kernel Methods in Deep Learning

By using the kernel methods to estimate the single-layer network's outputs for different samples, Neal (1996) find that such a network can become a Gaussian process (GP) if parameters are randomly initialized with a large width limit. Furthermore, recent researchers also extend these relations to neural networks with multiple layers (Lee et al., 2017; de G. Matthews et al., 2018) and other architectures (Novak et al., 2019; Garriga-Alonso et al., 2019). These works study the neural networks with parameters that are randomly initialized and fixed throughout the training process except for the last classification layer, which is named weakly-trained by Arora et al. (2019). Although these models are "weakly-trained", researchers can still obtain some useful conclusions for current neural networks with such weakly-trained models. For example, the related mean-field theory (Chen et al., 2018a; Gilboa et al., 2019; Hayou et al., 2019) explains the gradient vanishing and exploding phenomenon during the back-propagation which can also apply to other structures like CNNs, and RNNs. Meronen et al. (2020) studies the stationary kernels to enhance the neural networks' uncertainty calibration abilities by designing different activation functions.

Apart from the weakly trained models, Jacot et al. (2019); Arora et al. (2019) propose Neural Tangent Kernel (NTK) and its variants. They have proved that the sample kernel for infinite wide networks with proper initialization can converge to a static neural tangent kernel if their models are trained by gradient descent with infinitesimal steps (gradient flow). Then prediction can be made with NTK. Therefore, the NTK model is a theoretical model with strict constraints and its weights are not learnable. Although our model can also be regarded as an infinite wide model, we note there are many differences between ours and the above models. First, our models use the kernel method calculating on the input features and output features while the above uses the kernel method calculating on samples. Secondly, our model can be assumed as using the "weight-tied infinite wide" projection parameterized by the learnable parameter which can be updated during training. Thereby, there are many differences between our KerDEQ with NTKs and NTK-DEQ (Feng & Kolter, 2021) although there may exist some similar terms in our paper.

3 Gaussian Kernel Inspired OptEqs

3.1 Formulation and Structure of KerDEQ

Before starting our analysis, we need to reformulate the original formulations of OptEqs' equilibrium equation Eqn (1) and hidden optimization problem Eqn (2) for convenience. We replace $\tilde{\mathbf{W}}_c\tilde{\mathbf{W}}^\top$ with $\mathbf{W}_c$, $\boldsymbol{W} := \tilde{\boldsymbol{W}}^\top$, and optimize $\boldsymbol{z} := \tilde{\boldsymbol{W}}^{-1\top}\tilde{\boldsymbol{z}}$. Then the original OptEqs' optimization problem can be reformulated as:

$$\min_{\mathbf{z}} G(\mathbf{z};\mathbf{x}) = \min_{\mathbf{z}} \left[\mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|^2 - \langle \mathbf{U}\mathbf{x}+\mathbf{b}, \mathbf{z}\rangle - \frac{1}{2}\|\mathbf{W}\mathbf{z}\|^2\right], \tag{3}$$

and the reformulated equilibrium equation for OptEqs with input $\boldsymbol{x}$ can be obtained by reformulating Eqn (3)'s first order stationary condition as follows:

$$\mathbf{z}^* = \sigma\left(\mathbf{W}^\top\mathbf{W}\mathbf{z}^* + \mathbf{U}\mathbf{x} + \mathbf{b}\right), \tag{4}$$

where $\mathbf{U}, \mathbf{W}, \boldsymbol{b}$ are learnable parameters and f is the proximal term to constrain the features. Different choice of f can also induce different activation functions σ. If we choose the widely used ReLU function as the activation function, then $f(x) = \begin{cases} 0, & x \geq 0, \\ \infty, & x < 0. \end{cases}$ is the positive indicator function. From problem Eqn (3), OptEqs can be explained as extracting the features by minimizing the

similarity term with the input feature $\mathbf{Ux}+\mathbf{b}$ through linear kernel while regularizing the equilibrium state by the regularizer term $\mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|^2$. Such an explanation can also extend to other DEQs (Bai et al., 2019; Winston & Kolter, 2020) only if they constrain the weight parameter for the equilibrium states to be symmetric. We note that the symmetric constraints won't influence the final performance much as many works (Liu et al., 2021; Hu et al., 2019) show.

Thereby, a natural idea is whether we can use other kernel functions like the widely used Gaussian Kernels to extract the input features for the equilibrium state. Since the Gaussian kernel can naturally regularize $\mathbf{Wz}$'s norm, we can formulate the OptEqs' hidden optimization problem with Gaussian kernels as below:

$$\min_{\mathbf{z}} G(\mathbf{z};\mathbf{x}) = \mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|^2 - \frac{1}{2\gamma}\exp^{-\gamma\|(\mathbf{Ux}+\mathbf{b}-\mathbf{Wz})\|_F^2}, \tag{5}$$

where γ is the hyperparameter denoting the reciprocal of Gaussian kernels' variance for scaling. Then from the KKT condition, we can get the Gaussian Kernel method induced deep equilibrium model with Gaussian Kernel (KerDEQ) as the following fixed-point equation:

$$\mathbf{z} = \sigma\left[\exp^{-\gamma\|(\mathbf{Ux}+\mathbf{b}-\mathbf{Wz})\|_F^2}\mathbf{W}^\top(-\mathbf{Wz}+\mathbf{Ux}+\mathbf{b})\right] \tag{6}$$

Compared with linear kernels, Gaussian kernels can easily extract the non-linear relations from the input features and shows more stable and powerful performance in various machine learning approaches like SVM (Hu et al., 2019; Scholkopf et al., 1997) and others. Therefore, our KerDEQ is supposed to enjoy more representative abilities than the original OptEqs. In the following parts of this section, we will analyze the theoretical advantages of our KerDEQ against the vanilla OptEqs. And we'll also empirically evaluate the KerDEQ's performance in the next section.

3.2 Towards OptEqs with infinite width

Since the Gaussian kernel can be regarded as computing the similarities of samples by projecting them to the infinite-dimensional space, the first advantage of our KerDEQ is that our model can extract the input features from the infinite-dimension level. After projecting the input features $\boldsymbol{x}$ and output embedding $\boldsymbol{z}$ to the infinite-dimensional space by the "weight-tied" projection, our KerDEQ can be regarded calculate the similarity via the linear kernel like OptEqs in the infinite-dimension space as follows:

Theorem 1. *The KerDEQ with Gaussian kernels (Eqn (6)) is the same as dealing with the Optimized induced Equilibrium Models whose hidden optimization problem without the norm of* $\mathbf{Wz}$ *formulated as follows:*

$$\min_{\mathbf{z}} G(\mathbf{z};\mathbf{x}) = \mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|^2 - \lambda\left\langle \Phi_{\mathbf{U}}(\mathbf{x}+\mathbf{U}^{-1}\mathbf{b}), \Phi_{\mathbf{W}}(\mathbf{z})\right\rangle$$

where $\Phi_{\mathbf{W}}(\mathbf{z}) = \left[\mathbf{1}, \sqrt{2\gamma}\boldsymbol{\Phi}_{\mathbf{W}}^{(\mathbf{1})}(\mathbf{z}), ..., \sqrt{\frac{(2\gamma)^i}{i!}}\Phi_{\mathbf{W}}^{(i)}(\mathbf{z}), ...\right] \in \mathbb{R}^{1\times\infty}$ *which projects the features to the infinite-dimensional space. And* $\Phi_{\mathbf{W}}^{(i)} : \mathbb{R}^n \to \mathbb{R}^{in^i}$ *is the* k*-tuple permutation with repetitions formulated as follows:*

$$\Phi_{\mathbf{W}}^{(i)} = \left[\underbrace{\overbrace{(Wx)_0(Wx)_0...(Wx)_0}^{i}, \overbrace{(Wx)_0(Wx)_0...(Wx)_1}^{i}, ..., \overbrace{(Wx)_j(Wx)_k...(Wx)_m}^{i}, ...}_{in^i}\right] \tag{7}$$

where $(Wx)_j$ *denotes the* j*-th element of vector* $\mathbf{Wx}$*. Then the Gaussian kernel can also be regarded as calculating linear kernels after the weight-tied infinite wide projection* $\Phi_{\mathbf{W}}$ *and* $\Phi_{\mathbf{U}}$*.*

From the above analysis, one can see that our KerDEQ uses the linear kernels to extract features after projecting the input features and output features to the infinite-dimensional space with the weight-tied infinite wide projection $\Phi_{\mathbf{W}}$ and $\Phi_{\mathbf{U}}$. Compared with OptEqs which do the same operation in low-dimensional space, features in high-dimensional space are more easily to be extracted. Therefore, our model can show better performance as the following experiments show.

Apart from the above analysis, we are trying to analyze our the Gaussian kernel's generalization bound against the original OptEqs. Let $f_{KerDEQ}^c = \mathbf{W}_c f_{KerDEQ}(\mathbf{x}) + \mathbf{b}_c$ with $\mathbf{W}_c$ and $\mathbf{b}_c$ are

weights for classifiers. We have $L_\eta(f^c_{KerDEQ})$ denote the expected margin loss at margin η of our KerDEQ on the data distribution $\mathcal{D}$ defined as follows,

$$L_\eta(f^c_{KerDEQ}) = \mathbb{P}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\left[f^c_{KerDEQ}(\mathbf{x})_y \leq \eta + \max_{j\neq y} f^c_{KerDEQ}(\mathbf{x})_j\right] \tag{8}$$

and $\hat{L}_\eta(f^c_{KerDEQ})$ denote the empirical margin loss on the training set. Then we can analyze the generalization bound for our KerDEQ following Pabbaraju et al. (2021)'s settings.

Theorem 2. *For any $\delta, \eta > 0$, with probability at least $1-\delta$ over the training set of size M, input $\|\mathbf{x}\|_2$ is bounded by B, $\mu := \max\{\|\mathbf{U}\|_2, \|\mathbf{W}\|_2, \|\mathbf{W}_c\|_2, \|\mathbf{b}\|_2\} < 1$ and with proper choice of γ,*

$$L_0(f^c_{KerDEQ}) \leq \hat{L}_\eta(f^c_{KerDEQ}) + \mathcal{O}\left(\sqrt{\frac{h\ln(h)\left[(\beta_{\max}\mu)\mu^3 B + (2\mu\beta_{\max}+1)(1-\beta_{\max}m)\mu B + (1-\beta_{max}m)^2)^2\right]^2\mathcal{B}_W}{\eta^2(1-\beta_{\max}m)^4 M}} + \frac{\ln(\frac{M\sqrt{M}}{\delta})}{M}\right),$$

$$L_0(f^c_{opteq}) \leq \hat{L}_\eta(f^c_{opteq}) + \mathcal{O}\left(\sqrt{\frac{h\ln(h)\left[\mu^3 B + (1-m)\mu B + (1-m)^2\right]^2\mathcal{B}_W}{\eta^2(1-m)^4 M}} + \frac{\ln(\frac{M\sqrt{M}}{\delta})}{M}\right),$$

where $\mathcal{B}_W := \|\boldsymbol{W}^\top\boldsymbol{W}\|_F^2 + \|\mathbf{U}\|_F^2 + \|\mathbf{b}\|_F^2 + \|\mathbf{W}_c\|_F^2 + \|\mathbf{b}_c\|_F^2$, the maximum scaling number is defined by $\beta_{max} := \max_{\mathbf{x}\in\mathcal{D}} \exp^{-\gamma\|\mathbf{U}\mathbf{x}+\mathbf{b}-\mathbf{W}\mathbf{z}\|_F^2} < 1$ and $m = \|\boldsymbol{W}^\top\boldsymbol{W}\|_2$ which is less than 1.

Remark 1. *Assuming $\|\boldsymbol{W}^\top\boldsymbol{W}\|_2 = 0.9$ and $\mu = 0.9$, then we can get $\frac{\beta_{\max}\mu}{1-\beta_{\max}m} < \frac{1}{1-m}$ and $\frac{2\mu\beta_{\max}+1}{1-\beta_{\max}m} \leq \frac{1}{1-m}$ only if $\beta_{\max} < 0.83$. In the meanwhile, our KerDEQ's generalization bound is tighter than the original OptEq.*

From the theorem, one can see that our KerDEQ's generalization bound can be tighter than the original OptEqs with a proper choice of γ. Therefore, our KerDEQ can show better performance.

3.3 Towards Better Uncertainty Calibration Ability.

The model's uncertainty calibration ability trying to evaluate whether the models can not only give the right predictions but also can return better confident scores on samples. For example, we wish a model with better uncertain calibration ability can give correct predictions with high confidence on the learned samples. And on unseen samples, models cannot be over-confident, which means they cannot return high prediction scores on wrong labels Guo et al. (2017). If a model can achieve the above statements, it will be more reliable since its high prediction score means that the model's prediction is much more likely to be seen and correctly classified. Researchers can analyze such properties from the view of sample similarities when the weight parameters go to infinite dimension and obey the spherical distributions as former works (Meronen et al., 2020; de G. Matthews et al., 2018). With the above distribution, the similarity κ for samples $\mathbf{x}, \mathbf{y}$ can be defined as follows:

$$\kappa(\mathbf{x},\mathbf{y}) = \mathbb{E}\left[f(\mathbf{x})^\top f(\mathbf{y})\right] = \int_{\mathbb{R}} f_{\mathbf{w}}(\mathbf{x})^\top f_{\mathbf{w}}(\mathbf{y}) p(\mathbf{w}) d\mathbf{w}, \tag{9}$$

with $p(\mathbf{w})$ is the distribution of weight. Then if κ is dependent on the samples distance, it can be more reliable Meronen et al. (2020). Since if the samples has been seen in the training set, model can give the similar outputs to the neighboring samples in the test set. And the output will be different for the far away samples in the test set like random guess. In this section, we are going to analyze such properties of our KerDEQ and OptEqs by analyzing sample output similarities. The main conclusion is summarized as follows.

Theorem 3. *Assuming the inputs are bounded by B, the weight $\mathbf{W}$'s ℓ_2 are bounded by $\mu < 1$ to ensure the convergence of the equilibrium and $\mathbf{U}$ obeys the spherical Gaussian distributions $\mathcal{N}(0, \mathbb{E}[U_i^2]\boldsymbol{I})$. Then we have the following conclusions for the expectation of the similarity for $\mathbf{x}, \mathbf{y}$ defined in Eqn (9) is,*

$$\kappa_{kerdeq}(\boldsymbol{x},\boldsymbol{y}) \leq \frac{\mu^2 D \exp^{-\gamma(\|\boldsymbol{U}\|_2\|\boldsymbol{x}-\boldsymbol{y}\|_F^2)}\mathbb{E}[U_i^2]\|\mathbf{x}\|\|\mathbf{y}\|\left(\sin\theta_0 + (\pi-\theta_0)\cos\theta_0\right)}{2\pi(1-\beta_{\max}\mu^2)^2}, \tag{10}$$

$$\kappa_{opteq}(\boldsymbol{x},\boldsymbol{y}) \leq \frac{\mathbb{E}[U_i^2]\|\mathbf{x}\|\|\mathbf{y}\|\left(\sin\theta_0 + (\pi-\theta_0)\cos\theta_0\right)}{2\pi(1-\mu^2)^2}, \tag{11}$$

where $D = \exp^{\gamma\|\mathbf{W}\|_2 B}$ and $\beta_{max} := \max_{\mathbf{x}\in\mathcal{D}} \exp^{-\gamma\|\mathbf{U}\mathbf{x}-\mathbf{W}\mathbf{z}\|_F^2} < 1$. $\theta_0 = \cos^{-1}(\frac{\langle\mathbf{x},\mathbf{y}\rangle}{\|\mathbf{x}\|\|\mathbf{y}\|})$ is defined as the angle between the samples.

From the above results, the traditional OptEqs' Gram matrix for samples' outputs only depends on the samples' norms and angles. However, if the samples are far away but in the same direction, the similarity of the samples is still large. Therefore, OptEqs are more likely to be over-confident on these unseen samples. However, our KerDEQ's Gram matrix for outputs is also bounded by the distance of the sample. The similarity for the output features of our KerDEQ is smaller if the samples are far away, which are more likely to be unseen samples. Furthermore, the output similarity for our KerDEQ's output can also be large if the samples are near enough. Therefore, our KerDEQ can enjoy better uncertainty calibration ability compared with the original OptEqs.

3.4 Patch Splitting in KerDEQ.

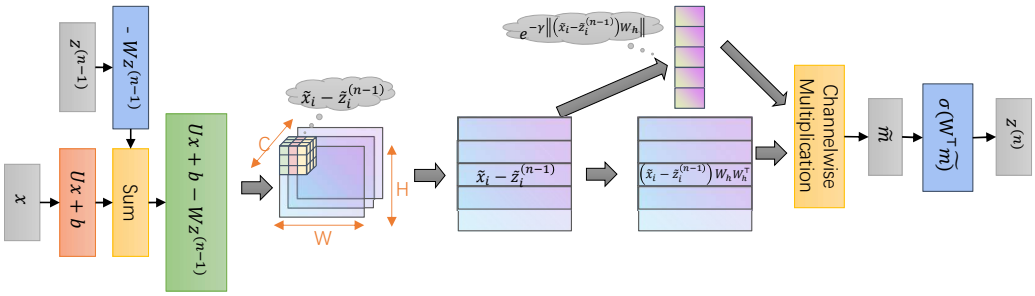


Figure 1: The sketch map of one layer KerDEQ's n-th fixed point iteration. $\mathbf{x}$ is input and $\mathbf{z}^{(n-1)}, \mathbf{z}^{(n)}$ are the output of $(n-1)$-th, n-th iteration. $\tilde{z}, \tilde{x}$ and $\tilde{m}$ are defined in Algorithm 1

.

Since different parts of images have different impacts on the image classification, calculating the whole similarity using Gaussian kernels for KerDEQ is not enough. Inspired by Trockman & Kolter (2022), we also split the feature map $\mathbf{Ux}$ into patches, and then our optimization problem becomes:

$$\min_{\mathbf{z}} G(\mathbf{z};\mathbf{x}) = \mathbf{1}^\top \tilde{\sigma}(\mathbf{z}) - \frac{1}{2\gamma}\sum_{i=1}^{N} \exp^{-\gamma\|(\tilde{\mathbf{x}}_i-\tilde{\mathbf{z}}_i)\mathbf{W}_h\|_F^2} \tag{12}$$

where $\tilde{\mathbf{x}}_i \in \mathbb{R}^{c_s p^2}$ is the i-th patch of $\mathbf{Ux}+\mathbf{b}$ while $\tilde{\mathbf{z}}_i \in \mathbb{R}^{c_s p^2}$ is the i-th patch of $\mathbf{Wz}$ and $\mathbf{W}_h \in \mathbb{R}^{c_s p^2 \times c_{hid}}$ is a linear layer to project patches with different size to the constant dimension. c_s denotes the channel splitting number, p denotes the patch size, and c_{hid} denotes the hidden dimension number of patches after linear projection. Then the i-th fixed point iteration corresponding to KerDEQ's hidden optimization problem can be formulated as Alg 1. Since the original OptEqs use linear kernels, the structure will be the same no matter using the patch splitting algorithms or not. Therefore, the patch-splitting approach is unique to our KerDEQ.

Algorithm 1: Calculating one layer KerDEQ.

Require: $\mathbf{Wz}^{(i)}, \mathbf{Ux} \in \mathbb{R}^{c\times h\times w}$, channel split c_s, patch size p, hidden layer $\mathbf{W}_h \in \mathbb{R}^{c_s p^2 \times 32}$

Ensure: Get the output $\mathbf{z}^{(i+1)}$ of i-th fixed point iteration.

Rearrange $\mathbf{Wz}^{(i)} \to \tilde{\mathbf{z}}$, $\mathbf{Ux}+\mathbf{b} \to \tilde{\mathbf{x}} \in \mathbb{R}^{\frac{chw}{c_s p^2}\times(c_s p^2)}$

$\tilde{\mathbf{m}} = diag\left(\exp^{-\gamma\|(\tilde{\mathbf{x}}_j-\tilde{\mathbf{z}}_j)\mathbf{W}_h\|_F^2}\right)(\tilde{\mathbf{x}}-\tilde{\mathbf{z}}_j)\,\mathbf{W}_h\mathbf{W}_h^\top$

Rearrange $\tilde{\mathbf{m}} \to \mathbf{m} \in \mathbb{R}^{c\times h\times w}$

return $\mathbf{z}^{(i+1)} = \sigma\left(\mathbf{W}^\top \mathbf{m}\right)$

4 Empirical Results

4.1 Experiment Settings

Firstly, we finish the experiments on the widely used image classification datasets. Then we also compare their performance with the designed experiments for the OOD calibration test. Furthermore, we finish the experiments to show the robustness of our KerDEQ compared with other equi-

librium models and ResNet (He et al., 2016). The ablation studies on other hyperparameters are also conducted in this section.

In the experiments, we parallel several KerDEQ with different input scales and we average each branch's output after average pooling or nearest up-sampling for branches' fusion. We use the Anderson algorithms to obey the forward equilibrium like other implicit models Li et al. (2021); Xie et al. (2021) and we use the Phantom gradient Geng et al. (2021) for the back-propagation. The models are trained by SGD with the step learning rate schedule. Our experiments are finished on PyTorch platform Paszke et al. (2019) with RTX-3090. Details can be found in the Appendix.

4.2 Results for Image Classification

Firstly, we finish the experiments on CIFAR-10 and CIFAR-100 which are widely used datasets for image classification for small images. In the experiment, we parallel 6 branches KerDEQ with the input scale is $32, 16, 8, 8, 4, 4$. The details can be found in the Appendix. As for the comparison, we also conduct the experiments of the same training procedure for MDEQ, MOptEqs, and ResNet. The results are listed in Table 1.

	Model Size	Accuracy
ResNet-18	10M	$93.5 \pm 0.2\%$
MDEQ	10M	$94.2 \pm 0.3\%$
MOptEq	8M	$94.6 \pm 0.2\%$
KerDEQ	5M	$94.7 \pm 0.1\%$
KerDEQ	8M	$\mathbf{95.3 \pm 0.2}\%$

(a) CIFAR-10.

	Model Size	Accuracy
ResNet-18	10M	$74.5 \pm 0.2\%$
MDEQ	10M	$74.7 \pm 0.3\%$
MOptEq	8M	$75.6 \pm 0.2\%$
KerDEQ	5M	$76.5 \pm 0.3\%$
KerDEQ	8M	$\mathbf{77.8 \pm 0.2}\%$

(b) CIFAR-100.

Table 1: The Empirical Results for image classification on CIFAR-10 and CIFAR-100.

From the above experiments, one can see that our KerDEQ enjoys clear advantages on the CIFAR dataset, which demonstrates the powerful generalization ability of other models no matter implicit models or explicit models.

	Model Size	Accuracy
ResNet-18	12M	$92.3 \pm 0.1\%$
MDEQ	10M	$91.5 \pm 0.2\%$
MOptEq	10M	$92.4 \pm 0.2\%$
KerDEQ	6M	$\mathbf{92.9 \pm 0.2}\%$
KerDEQ	13M	$\mathbf{93.2 \pm 0.1}\%$

(b) Empirical Results on ImageNette.

	Model Size	Accuracy
ResNet-18	11M	80.9%
ResNet-50	23M	81.7%
MDEQ	10M	81.3%
MOptEq	13M	81.5%
KerDEQ	6M	$\mathbf{82.2}\%$
KerDEQ	13M	$\mathbf{83.9}\%$

(b) Empirical Results on ImageNet-100

Table 2: The Empirical Results for image classification on ImageNette and ImageNet-100.

Apart from the small datasets, we also conduct experiments on the large-scale images listed in Table 2. From the table, one can see that our KerDEQ can outperform other models with clear advantages. The results demonstrate the superiority of our KerDEQ against others is consistent on large-scale inputs.

4.3 OOD Uncertainty Calibration Experiments

Models with better uncertainty calibration performance can predict the learned samples correctly with high confidence which will return low confidence scores on unseen samples. Thereby, we can design experiments on CIFAR datasets to evaluate the models' uncertainty calibration abilities based on other works Meronen et al. (2020) setting. Firstly, we train our KerDEQs and other baseline models for comparison only using the first five classes of the training set. Therefore, the first five classes in the test set can be regarded as the learned samples while the last five classes in the test set are the unseen samples which are also called the out-of-distribution (OOD) samples.

Then we can use the KL-divergence to evaluate the calibration abilities of different models. The KL-divergence is the widely used metric to evaluate p's similarities of distribution q defined as:

$$KL(p,q) = \sum_i p_i \log p_i - p_i \log q_i. \tag{13}$$

For the known classes, $p = \mathbb{P}(\mathbf{y}|\mathbf{x})$ is the one-hot vector of sample $\mathbf{x}$'s label and q is the model's prediction score for input $\mathbf{x}$. Since for the unknown classes, we wish the model with better calibration abilities will output almost the same confidence on different classes. Thereby, we choose q to be the uniform distribution for each class of unseen samples. Then we evaluate the uncertainty calibration ability of one branch OptEq and our KerDEQ with the above metrics listed in Table 3.

	Accuracy	Known KL	Unknown KL
OptEq	90.56%	0.38	1.51
KerDEQ	92.42%	0.306	1.21

Table 3: Empirical Results on CIFAR-10 with 5 classes training for OOD uncertainty. Lower is the better.

From the tables, one can see that our model also enjoys better uncertainty calibration abilities than OptEqs whose hidden optimization problem used the linear kernel as Eqn (3). The results also confirm the analysis we conduct in Section 3.3.

4.4 Ablation Studies on Corrupted Datasets

In other kernel-related machine learning approaches Soman et al. (2009); Zhang et al. (2017), Gaussian kernels are usually more robust against various noises and corruptions than the traditional linear kernel. Therefore, we also compare the robustness of our KerDEQ, MOptEqs, and ResNet on the CIFAR-10 corruption dataset, which contains 19 common corruptions including image transformation, image blurring, weather, and digital noises on CIFAR's test datasets. The results are listed in Figure 2.

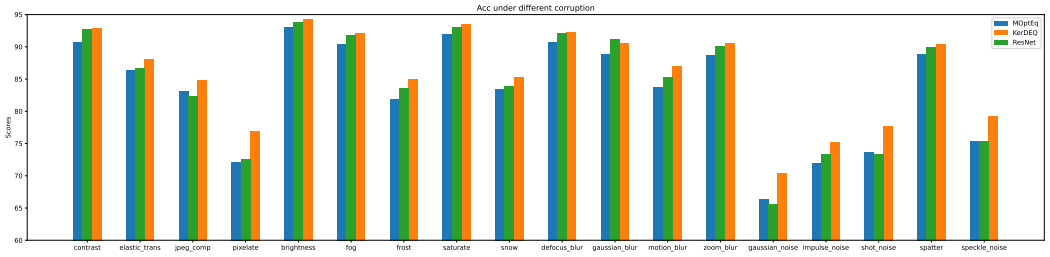


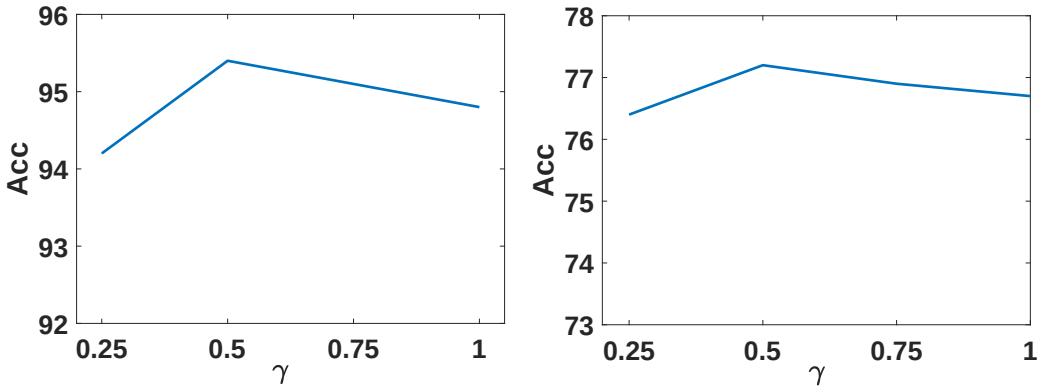
Figure 2: The results for different models under different corruptions.

From the result, one can see that our KerDEQ based on Gaussian Kernels is more robust than other implicit and explicit models. Especially, our KerDEQ can show better performance against the structured noise and some image transformation.

4.5 Ablation Studies on γ

The choice of γ will influence the activation of the output scale of the KerDEQ model. Too small γ will make the scaling part $\exp^{-\gamma\|\mathbf{U}\mathbf{x}+\mathbf{b}-\mathbf{W}\mathbf{z}\|_F^2}$ almost equal to 1, then the model will perform just the same as the original OptEqs using the linear Kernel. Too large γ will make the scaling term too small in most cases. Then the norm of the output $\mathbf{z}$ may converge around 0 and such a phenomenon may harm the representative ability. In order to explore the influence of γ, we finish the experiments on CIFAR-10 and CIFAR-100 with different γ's choices drawn in Figure 3.

From the figure, one can see that the accuracy will first raise and then decrease with the increase of the hyperparameter γ. 0.5 is a good choice for both models. Therefore, we choose $\gamma = 0.5$ in our experiments.

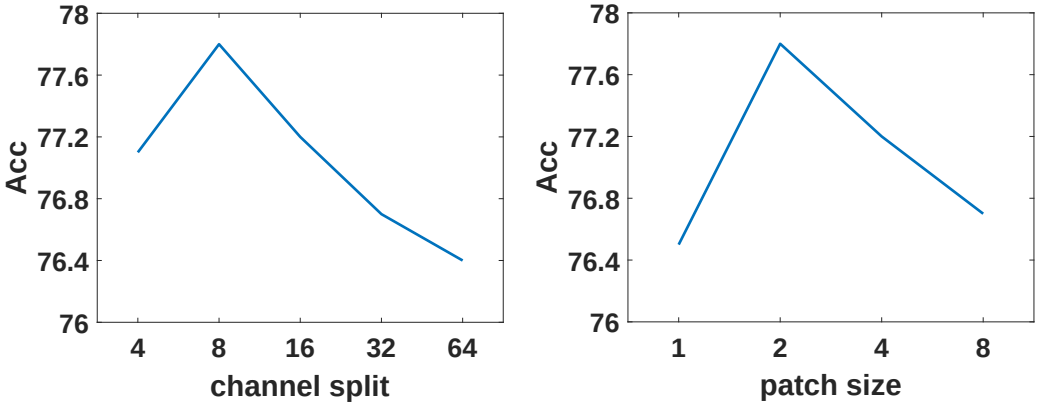


(a) CIFAR-10 (b) CIFAR-100

Figure 3: The influence of hyperparameter γ for our KerDEQ on CIFAR datasets.

4.6 Ablation Studies on Patch splitting

The channel splitting parameter and the patch size of our KerDEQ also will influence the performance of our model. If the choices are too large, then the kernel will only focus on the global information while the kernel may focus on local features if the channel splitting parameter and patch size are chosen to be too small. Due to this reason, we also conduct experiments trying to find their influence on the performance shown in Figure 4.



(a) Influence of the channel splitting parameter. (b) Influence of the patch size parameter.

Figure 4: The influence on the patch size and the channel splitting parameter for our KerDEQ on CIFAR-100 datasets.

From the figure, one can see that the accuracy will first raise and then decrease with the increase of the channel split and patch size. According to the empirical results, we choose patch size equals to 2 with channel split equals to 8 for CIFAR in the experiments . As for ImageNette and ImageNet-100 we choose patch size equals to 4 and channel split equals to 4.

5 Conclusions

In this paper, we propose a novel optimization induced deep equilibrium model with Gaussian kernels called KerDEQ. Our KerDEQ can be viewed as a model with weight-tied infinite width and infinite depth. Furthermore, we also theoretically analyzed our models' superiority in the model's generalization abilities and uncertainty calibration abilities against former OptEqs. Empirical results also demonstrate the effectiveness of our proposed models.

REFERENCES

Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pp. 136–145. PMLR, 2017.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in Neural Information Processing Systems*, 32, 2019.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32, 2019.

Shaojie Bai, Vladlen Koltun, and J Zico Kolter. Multiscale deep equilibrium models. *Advances in Neural Information Processing Systems*, 33:5238–5250, 2020.

Minmin Chen, Jeffrey Pennington, and Samuel Schoenholz. Dynamical isometry and a mean field theory of rnns: Gating enables signal propagation in recurrent neural networks. In *International Conference on Machine Learning*, pp. 873–882. PMLR, 2018a.

Qi Chen, Yifei Wang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Optimization-induced graph implicit nonlinear diffusion. In *International Conference on Machine Learning*, pp. 3648–3661. PMLR, 2022.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 6572–6583, 2018b.

KR1442 Chowdhary. Natural language processing. *Fundamentals of artificial intelligence*, pp. 603–649, 2020.

Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J Zico Kolter. End-to-end differentiable physics for learning and control. *Advances in neural information processing systems*, 31:7178–7189, 2018.

Alexander G. de G. Matthews, Jiri Hron, Mark Rowland, Richard E. Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1-nGgWC-.

Josip Djolonga and Andreas Krause. Differentiable learning of submodular models. *Advances in Neural Information Processing Systems*, 30:1013–1023, 2017.

Zhili Feng and J Zico Kolter. On the neural tangent kernel of equilibrium models, 2021. URL https://openreview.net/forum?id=8_7yhptEWD.

Adrià Garriga-Alonso, Carl Edward Rasmussen, and Laurence Aitchison. Deep convolutional networks as shallow gaussian processes. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bklfsi0cKm.

Zhengyang Geng, Xin-Yu Zhang, Shaojie Bai, Yisen Wang, and Zhouchen Lin. On training implicit models. *Advances in Neural Information Processing Systems*, 34:24247–24260, 2021.

Laurent El Ghaoui, Fangda Gu, Bertrand Travacca, and Armin Askari. Implicit deep learning. *CoRR*, abs/1908.06315, 2019.

Dar Gilboa, Bo Chang, Minmin Chen, Greg Yang, Samuel S Schoenholz, Ed H Chi, and Jeffrey Pennington. Dynamical isometry and a mean field theory of lstms and grus. *arXiv preprint arXiv:1901.08987*, 2019.

Stephen Gould, Richard Hartley, and Dylan Campbell. Deep declarative networks: A new hope. *CoRR*, abs/1909.04866, 2019.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. Mean-field behaviour of neural tangent kernel for deep neural networks. *arXiv preprint arXiv:1905.13654*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Shell Xu Hu, Sergey Zagoruyko, and Nikos Komodakis. Exploring weight symmetry in deep neural networks. *Computer Vision and Image Understanding*, 187:102786, 2019.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Freeze and chaos for dnns: an NTK view of batch normalization, checkerboard and boundary effects. *CoRR*, abs/1907.05715, 2019. URL http://arxiv.org/abs/1907.05715.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

Mingjie Li, Yisen Wang, Xingyu Xie, and Zhouchen Lin. Optimization inspired multi-branch equilibrium models. In *International Conference on Learning Representations*, 2021.

Guangcan Liu and Ping Li. Low-rank matrix completion in the presence of high coherence. *IEEE Transactions on Signal Processing*, 64(21):5623–5633, 2016.

Guangcan Liu, Shiyu Chang, and Yi Ma. Blind image deblurring using spectral properties of convolution operators. *IEEE Transactions on image processing*, 23(12):5047–5056, 2014.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Juncheng Liu, Kenji Kawaguchi, Bryan Hooi, Yiwei Wang, and Xiaokui Xiao. Eignn: Efficient infinite-depth graph neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pp. 689–696, 2009.

Lassi Meronen, Christabella Irwanto, and Arno Solin. Stationary activations for uncertainty calibration in deep learning. *Advances in Neural Information Processing Systems*, 33:2338–2350, 2020.

Radford M Neal. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pp. 29–53. Springer, 1996.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *CoRR*, abs/1707.09564, 2017. URL http://arxiv.org/abs/1707.09564.

Roman Novak, Lechao Xiao, Yasaman Bahri, Jaehoon Lee, Greg Yang, Jiri Hron, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=B1g30j0qF7.

Chirag Pabbaraju, Ezra Winston, and J Zico Kolter. Estimating lipschitz constants of monotone deep equilibrium models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=VcB4QkSfyO.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Yi-Ling Qiao, Junbang Liang, Vladlen Koltun, and Ming Lin. Scalable differentiable physics for learning and control. In *International Conference on Machine Learning*, pp. 7847–7856. PMLR, 2020.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Bernhard Scholkopf, Kah-Kay Sung, Christopher JC Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE transactions on Signal Processing*, 45(11):2758–2765, 1997.

KP Soman, R Loganathan, and V Ajay. *Machine learning with SVM and other kernel methods*. PHI Learning Pvt. Ltd., 2009.

Ivana Tošić and Pascal Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2): 27–38, 2011.

Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022.

Russell Tsuchida, Fred Roosta, and Marcus Gallagher. Invariance of weight distributions in rectified mlps. In *International Conference on Machine Learning*, pp. 4995–5004. PMLR, 2018.

Po-Wei Wang, Priya Donti, Bryan Wilder, and Zico Kolter. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *International Conference on Machine Learning*, pp. 6545–6554. PMLR, 2019.

Ezra Winston and J. Zico Kolter. Monotone operator equilibrium networks. *CoRR*, abs/2006.08591, 2020. URL `https://arxiv.org/abs/2006.08591`.

Xingyu Xie, Jianlong Wu, Guangcan Liu, Zhisheng Zhong, and Zhouchen Lin. Differentiable linearized admm. In *International Conference on Machine Learning*, pp. 6902–6911. PMLR, 2019.

Xingyu Xie, Jianlong Wu, Guangcan Liu, Zhisheng Zhong, and Zhouchen Lin. Optimization induced deep equilibrium networks. *arXiv preprint arXiv:2105.13228*, 2021.

Chong You, Daniel Robinson, and René Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3918–3927, 2016.

Hongyang Zhang, Zhouchen Lin, Chao Zhang, and Edward Y Chang. Exact recoverability of robust pca via outlier pursuit with tight recovery bounds. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

WeiChuan Zhang, YaLi Zhao, Toby P Breckon, and Long Chen. Noise robust image edge detection based upon the automatic anisotropic gaussian kernels. *Pattern Recognition*, 63:193–205, 2017.

Xiao Zhang, Lingxiao Wang, Yaodong Yu, and Quanquan Gu. A primal-dual analysis of global optimality in nonconvex low-rank matrix recovery. In *International conference on machine learning*, pp. 5862–5871. PMLR, 2018.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.