Explore Positive Noise in Deep Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

In computer vision, noise is conventionally viewed as a harmful perturbation in various deep learning architectures, such as convolutional neural networks (CNNs) and vision transformers (ViTs), as well as different tasks like image classification and transfer learning. However, this paper aims to rethink whether the conventional proposition always holds. We demonstrate that specific noise can boost the performance of various deep architectures under certain conditions. We theoretically prove the enhancement gained from positive noise by reducing the task complexity defined by information entropy and experimentally show the significant performance gain in large image datasets, such as the ImageNet. Herein, we use the information entropy to define the complexity of the task. We categorize the noise into two types, positive noise (PN) and harmful noise (HN), based on whether the noise can help reduce the complexity of the task. Extensive experiments of CNNs and ViTs have shown performance improvements by proactively injecting positive noise, where we achieve an unprecedented top 1 accuracy over 95% on ImageNet. Both theoretical analysis and empirical evidence have confirmed that the presence of positive noise, can benefit the learning process, while the traditionally perceived harmful noise indeed impairs deep learning models. The different roles of noise offer new explanations for deep models on specific tasks and provide a new paradigm for improving model performance. Moreover, it reminds us to utilize noise rather than suppress noise.

1 Introduction

Noise, conventionally regarded as a hurdle in machine learning and deep learning tasks, is universal and unavoidable due to various reasons, e.g., environmental factors, instrumental calibration, and human activities [23] [37]. In computer vision, noise can be generated from different phases: (1) Image Acquisition: Noise can arise from a camera sensor or other imaging device [33]. For example, electronic or thermal noise in the camera sensor can result in random pixel values or color variations that can be visible in the captured image. (2) Image Preprocessing: Noise can be introduced during preprocessing steps such as image resizing, filtering, or color space conversion [1]. For example, resizing an image can introduce aliasing artifacts, while filtering an image can result in the loss of detail and texture. (3) Feature Extraction: Feature extraction algorithms can be sensitive to noise in the input image, which can result in inaccurate or inconsistent feature representations [2]. For example, edge detection algorithms can be affected by noise in the image, resulting in false positives or negatives. (4) Algorithms: algorithms used for computer vision tasks, such as object detection or image segmentation, can also be sensitive to noise in the input data [6]. Noise can cause the algorithm to learn incorrect patterns or features, leading to poor performance.

Since noise is an unavoidable reality in engineering tasks, existing works usually make the assumption that noise has a consistently negative impact on the current task [30] [24]. Nevertheless, is the above assumption always valid? As such, it is crucial to address the question of whether noise can ever have a positive influence on deep learning models. This work aims to provide a comprehensive answer to this question, which is a pressing concern in the deep learning community. We recognize that the imprecise definition of noise is a critical factor leading to the uncertainties surrounding the identification and characterization of noise. To address these uncertainties, an in-depth analysis of the task's complexity is imperative for arriving at a rigorous answer. By using the definition of task entropy, it is possible to categorize noise into two distinct categories: positive noise (PN) and harmful noise (HN). PN decreases the complexity of the task, while HN increases it, aligning with the conventional understanding of noise.

1.1 Scope and Contribution

Our work aims to investigate how various types of noise affect deep learning models. Specifically, the study focuses on three common types of noise, i.e., Gaussian noise, linear transform noise, and salt-and-pepper noise. Gaussian noise refers to random fluctuations that follow a Gaussian distribution in pixel values at the image level or latent representations in latent space [29]. Linear transforms, on the other hand, refer to affine elementary matrix transformations to the dataset of original images or latent representations, where the elementary matrix is row equivalent to an identity matrix [36]. Salt-and-pepper noise is a kind of image distortion that adds random black or white values at the image level or to the latent representations [7].

This paper analyzes the impact of these types of noise on the performance of deep learning models for image classification and domain adaptation tasks. Two popular model families, Vision Transformers (ViTs) and Convolutional Neural Networks (CNNs), are considered in the study. Image classification is one of the most fundamental tasks in computer vision, where the goal is to predict the class label of an input image. Domain adaptation is a practically meaningful task where the training and test data come from different distributions, also known as different domains. By investigating the effects of different types of noise on ViTs and CNNs for typical deep learning tasks, the paper provides insights into the influences of noises on deep models. The findings presented in this paper hold practical significance for enhancing the performance of various types of deep learning models in real-world scenarios.

The contributions of this paper are summarized as follows:

- We re-examined the conventional view that noise, by default, has a negative impact on deep learning models. Our theoretical analysis and experimental results show that noise can be a positive support for deep learning models and tasks.
- We implemented extensive experiments with different deep models, such as CNNs and ViTs, and on different deep learning tasks. Empowered by positive noise, we achieved state-of-the-art (SOTA) results in all the experiments presented in this paper.
- Instead of operating on the image level, our injecting noise operations are performed in the latent space. We theoretically analyze the difference between injecting noise on the image level and in the latent space.
- The theory and framework of reducing task complexity via positive noise in this work can be applied to any deep learning architecture. There is great potential for exploring the application of positive noise in other deep-learning tasks beyond the image classification and domain adaptation tasks examined in this study.

1.2 Related Work

Positive Noise In fact, within the signal-processing society, it has been demonstrated that random noise helps stochastic resonance improve the detection of weak signals [4]. Noises can have positive support and contribute to less mean square error compared to the best linear unbiased estimator when the mixing probability distribution is not in the extreme region [28]. Also, it has been reported that noise could increase the model generalization in natural language processing (NLP) [27]. Recently, the perturbation, a special case of positive noise, has been effectively utilized to implement self-refinement in domain adaptation and achieved state-of-the-art performance [36]. The latest research shows that by proactively adding specific noise to partial datasets, various tasks can benefit from the positive noise [19]. Besides, noises are found to be able to boost brain power and be useful in many neuroscience studies [21] [22].

Deep Model Convolutional Neural Networks have been widely used for image classification, object detection, and segmentation tasks, and have achieved impressive results [18][15]. However, these networks have limitations in terms of their ability to capture long-range dependencies and extract global features from images. Recently, Vision Transformers has been proposed as an alternative to CNNs [13]. ViT relies on self-attention mechanisms and a transformer-based architecture to enable global feature extraction and modeling of long-range dependencies in images [40]. The attention mechanism allows the model to focus on the most informative features of the input image, while the transformer architecture facilitates information exchange between different parts of the image. ViT has demonstrated impressive performance on a range of image classification tasks and has the potential to outperform traditional CNN-based approaches. However, ViT currently requires a large number of parameters and training data to achieve state-of-the-art results, making it challenging to implement in certain settings [45].

2 Preliminary

In information theory, the entropy [32] of a random variable x is defined as:

$$H(x) = \begin{cases} -\int p(x) \log p(x) dx & \text{if } x \text{ is continuous} \\ -\sum_x p(x) \log p(x) & \text{if } x \text{ is discrete} \end{cases} \tag{1}$$

where $p(x)$ is the distribution of the given variable x. And the mutual information (MI) of two random discrete variables (x, y) is denoted as [8]:

$$\begin{aligned} MI(x, y) =& D_{KL}(p(x, y) \parallel p(x) \otimes p(y)) \\ =& H(x) - H(x|y) \end{aligned} \tag{2}$$

where D_{KL} is the Kullback–Leibler divergence [16], and $p(x, y)$ is the joint distribution. The conditional entropy is defined as:

$$H(x|y) = -\sum p(x, y) \log p(x|y) \tag{3}$$

The above definitions can be readily expanded to encompass continuous variables through the substitution of the sum operator with the integral symbol. In this work, the noise is denoted by $\boldsymbol{\epsilon}$ if without any specific statement.

Before delving into the correlation between task and noise, it is imperative to address the initial crucial query of the mathematical measurement of a task $\mathcal{T}$. With the assistance of information theory, the complexity associated with a given task $\mathcal{T}$ can be measured in terms of the entropy of $\mathcal{T}$. Therefore, we can borrow the concepts of information entropy to explain the difficulty of the task. For example, a smaller $H(\mathcal{T})$ means an easier task and vice versa.

Since the entropy of task $\mathcal{T}$ is formulated, it is not difficult to define the mutual information of task $\mathcal{T}$ and noise $\boldsymbol{\epsilon}$,

$$MI(\mathcal{T}, \boldsymbol{\epsilon}) = H(\mathcal{T}) - H(\mathcal{T}|\boldsymbol{\epsilon}) \tag{4}$$

Formally, if the noise can help reduce the complexity of the task, i.e., $H(\mathcal{T}|\boldsymbol{\epsilon}) < H(\mathcal{T})$ then the noise has positive support. Therefore, a noise $\boldsymbol{\epsilon}$ is defined as **positive noise** (PN) when the noise satisfies $MI(\mathcal{T}, \boldsymbol{\epsilon}) > 0$. On the contrary, when $MI(\mathcal{T}, \boldsymbol{\epsilon}) \leq 0$, the noise is considered as the conventional noise and named **harmful noise** (HN). The positive noise can be perceived as an augmentation of information gain brought by $\boldsymbol{\epsilon}$.

$$\begin{cases} MI(\mathcal{T}, \boldsymbol{\epsilon}) > 0 & \epsilon \text{ is positive noise} \\ MI(\mathcal{T}, \boldsymbol{\epsilon}) \leq 0 & \epsilon \text{ is harmful noise} \end{cases} \tag{5}$$

Moderate Model Assumption: The positive noise may not work for deep models with severe problems. For example, the model is severely overfitting where models begin to memorize the random fluctuations in the data instead of learning the underlying patterns. In that case, the presence of positive noise will not have significant positive support in improving the models' performance. Besides, when the models are corrupted under brute force attack, the positive noise also can not work.

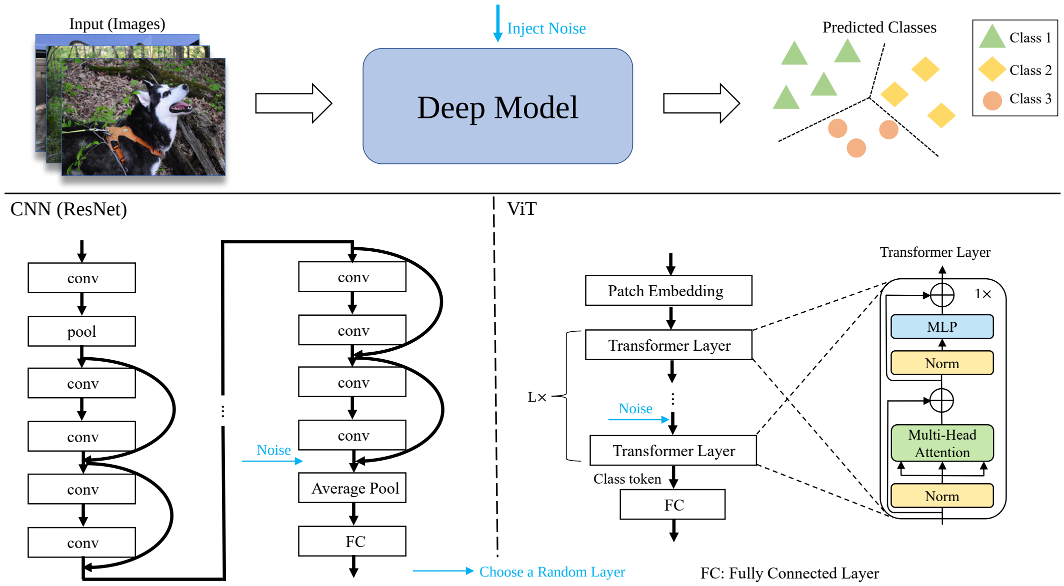


Figure 1: An overview of the proposed method. Above the black line is the standard pipeline for image classification. The deep model can be CNNs or ViTs. The noise is injected into a randomly chosen layer of the model represented by the blue arrow.

3 Methods

The idea of exploring the influence of noise on the deep models is straightforward. The framework is depicted in Fig. 1. This is a universal framework where there are different options for deep models, such as CNNs and ViTs. Through the simple operation of injecting noise into a randomly selected layer, a model has the potential to gain additional information to reduce task complexity, thereby improving its performance. It is sufficient to inject noise into a single layer instead of multiple layers since it imposes a regularization on multiple layers simultaneously.

For a classification problem, the dataset $(\boldsymbol{X}, \boldsymbol{Y})$ can be regarded as samplings derived from $D_{\mathcal{X},\mathcal{Y}}$, where $D_{\mathcal{X},\mathcal{Y}}$ is some unknown joint distribution of data points and labels from feasible space $\mathcal{X}$ and $\mathcal{Y}$, i.e., $(\boldsymbol{X}, \boldsymbol{Y}) \sim D_{\mathcal{X},\mathcal{Y}}$ [31]. Hence, given a set of k data points $\boldsymbol{X} = \{X_1, X_2, ..., X_k\}$, the label set $\boldsymbol{Y} = \{Y_1, Y_2, ..., Y_k\}$ is regarded as sampling from $\boldsymbol{Y} \sim D_{\mathcal{Y}|\mathcal{X}}$. The complexity of $\mathcal{T}$ on dataset $\boldsymbol{X}$ is formulated as [19]:

$$H(\mathcal{T}; \boldsymbol{X}) = -\sum_{\boldsymbol{Y} \in \mathcal{Y}} p(\boldsymbol{Y}|\boldsymbol{X}) \log p(\boldsymbol{Y}|\boldsymbol{X}) \tag{6}$$

The operation of adding noise at the image level can be formulated as:

$$\begin{cases} H(\mathcal{T}; \boldsymbol{X} + \boldsymbol{\epsilon}) = -\sum_{\boldsymbol{Y} \in \mathcal{Y}} p(\boldsymbol{Y}|\boldsymbol{X} + \boldsymbol{\epsilon}) \log p(\boldsymbol{Y}|\boldsymbol{X} + \boldsymbol{\epsilon}) & \boldsymbol{\epsilon} \text{ is additive noise} \\ H(\mathcal{T}; \boldsymbol{X}\boldsymbol{\epsilon}) = -\sum_{\boldsymbol{Y} \in \mathcal{Y}} p(\boldsymbol{Y}|\boldsymbol{X}\boldsymbol{\epsilon}) \log p(\boldsymbol{Y}|\boldsymbol{X}\boldsymbol{\epsilon}) & \boldsymbol{\epsilon} \text{ is multiplicative noise} \end{cases} \tag{7}$$

While the operation of proactively injecting noise in the latent space can be formulated as:

$$\begin{cases} H(\mathcal{T}; \boldsymbol{X} + \boldsymbol{\epsilon}) \overset{\star}{=} H(\boldsymbol{Y}; \boldsymbol{X} + \boldsymbol{\epsilon}) - H(\boldsymbol{X}) & \boldsymbol{\epsilon} \text{ is additive noise} \\ H(\mathcal{T}; \boldsymbol{X}\boldsymbol{\epsilon}) \overset{\star}{=} H(\boldsymbol{Y}; \boldsymbol{X}\boldsymbol{\epsilon}) - H(\boldsymbol{X}) & \boldsymbol{\epsilon} \text{ is multiplicative noise} \end{cases} \tag{8}$$

Step $\star$ differs from the conventional definition of conditional entropy, as our method injects the noise into the latent representations instead of the original images. The Gaussian noise is additive, the linear transform noise is also additive, and the salt-and-pepper is a multiplicative noise.

Gaussian Noise The Gaussian noise is one of the most common additive noises that appeared in computer vision tasks. The Gaussian noise is independent and stochastic, obeying the Gaussian distribution. Without loss of generality, defined as $\mathcal{N}(\mu, \sigma^2)$. Since our injection happens in the latent space, therefore, the complexity of the task is:

$$H(\mathcal{T}; \boldsymbol{X} + \boldsymbol{\epsilon}) \overset{\star}{=} H(\boldsymbol{Y}; \boldsymbol{X} + \boldsymbol{\epsilon}) - H(\boldsymbol{X}). \tag{9}$$

According to the definition in Equation 4, and making the distribution of $\boldsymbol{X}$ and $\boldsymbol{Y}$ multivariate normal distribution [5] [14], the mutual information with Gaussian noise is:

$$\begin{aligned} MI(\mathcal{T},\boldsymbol{\epsilon}) =& H(\boldsymbol{Y};\boldsymbol{X}) - H(\boldsymbol{X}) - (H(\boldsymbol{Y};\boldsymbol{X}+\boldsymbol{\epsilon}) - H(\boldsymbol{X})) \\ =& H(\boldsymbol{Y};\boldsymbol{X}) - H(\boldsymbol{Y};\boldsymbol{X}+\boldsymbol{\epsilon}) \\ =& \frac{1}{2}\log\frac{|\boldsymbol{\Sigma_X}||\boldsymbol{\Sigma_Y} - \boldsymbol{\Sigma_{YX}}\boldsymbol{\Sigma_X^{-1}}\boldsymbol{\Sigma_{XY}}|}{|\boldsymbol{\Sigma_{X+\epsilon}}||\boldsymbol{\Sigma_Y} - \boldsymbol{\Sigma_{YX}}\boldsymbol{\Sigma_{X+\epsilon}^{-1}}\boldsymbol{\Sigma_{XY}}|} \\ =& \frac{1}{2}\log\frac{1}{(1+\sigma_\epsilon^2\sum_{i=1}^k\frac{1}{\sigma_{X_i}^2})(1+\lambda\sum_{i=1}^k\frac{\text{cov}^2(X_i,Y_i)}{\sigma_{X_i}^2(\sigma_{X_i}^2\sigma_{Y_i}^2-\text{cov}^2(X_i,Y_i))})} \end{aligned} \tag{10}$$

where $\lambda = \frac{\sigma_\epsilon^2}{1+\sum_{i=1}^k\frac{1}{\sigma_{X_i}^2}}$, σ_ϵ^2 is the variance of the Gaussian noise, $\text{cov}(X_i, Y_i)$ is the covariance of sample pair X_i, Y_i, $\sigma_{X_i}^2$ and $\sigma_{Y_i}^2$ are the variance of data sample X_i and data label Y_i, respectively. The detailed derivations can be found in section 1.1.2 of the supplementary. Given a dataset, the variance of the Gaussian noise, and statistical properties of data samples and labels control the mutual information, we define the function:

$$\begin{aligned} f(\sigma_\epsilon^2) =& 1 - (1+\sigma_\epsilon^2\textstyle\sum_{i=1}^k\frac{1}{\sigma_{X_i}^2})(1+\lambda\displaystyle\sum_{i=1}^k\frac{\text{cov}^2(X_i,Y_i)}{\sigma_{X_i}^2(\sigma_{X_i}^2\sigma_{Y_i}^2-\text{cov}^2(X_i,Y_i))}) \\ =& -\sigma_\epsilon^2\textstyle\sum_{i=1}^k\frac{1}{\sigma_{X_i}^2} - \sigma_\epsilon^2\sum_{i=1}^k\frac{1}{\sigma_{X_i}^2}\cdot\lambda\displaystyle\sum_{i=1}^k\frac{\text{cov}^2(X_i,Y_i)}{\sigma_{X_i}^2(\sigma_{X_i}^2\sigma_{Y_i}^2-\text{cov}^2(X_i,Y_i))} - \lambda\sum_{i=1}^k\frac{\text{cov}^2(X_i,Y_i)}{\sigma_{X_i}^2(\sigma_{X_i}^2\sigma_{Y_i}^2-\text{cov}^2(X_i,Y_i))} \end{aligned} \tag{11}$$

Since $\epsilon^2 \geq 0$ and $\lambda \geq 0$, $\sigma_{X_i}^2\sigma_{Y_i}^2 - \text{cov}^2(X_i,Y_i) = \sigma_{X_i}^2\sigma_{Y_i}^2(1-\rho_{X_iY_i}^2) \geq 0$, where $\rho_{X_iY_i}$ is the correlation coefficient, the sign of $f(\sigma_\epsilon^2)$ is negative. We can conclude that Gaussian noise injected into the latent space is harmful to the task. More details and the Gaussian noise added to the image level are provided in the supplementary.

Linear Transform Noise This type of noise is obtained by elementary transformation of the features matrix, i.e., $\boldsymbol{\epsilon} = Q\boldsymbol{X}$, where Q is an elementary matrix. We name the Q the quality matrix since it controls the property of linear transform noise and determines whether positive or harmful. In the linear transform noise injection in the latent space case, the complexity of the task is:

$$H(\mathcal{T};\boldsymbol{X}+Q\boldsymbol{X}) \stackrel{\star}{=} H(\boldsymbol{Y};\boldsymbol{X}+Q\boldsymbol{X}) - H(\boldsymbol{X}) \tag{12}$$

The mutual information is then formulated as:

$$\begin{aligned} MI(\mathcal{T},Q\boldsymbol{X}) \stackrel{\star}{=}& H(\boldsymbol{Y};\boldsymbol{X}) - H(\boldsymbol{X}) - (H(\boldsymbol{Y};\boldsymbol{X}+Q\boldsymbol{X}) - H(\boldsymbol{X})) \\ =& H(\boldsymbol{Y};\boldsymbol{X}) - H(\boldsymbol{Y};\boldsymbol{X}+Q\boldsymbol{X}) \\ =& \frac{1}{2}\log\frac{|\boldsymbol{\Sigma_X}||\boldsymbol{\Sigma_Y} - \boldsymbol{\Sigma_{YX}}\boldsymbol{\Sigma_X^{-1}}\boldsymbol{\Sigma_{XY}}|}{|\boldsymbol{\Sigma}_{(I+Q)\boldsymbol{X}}||\boldsymbol{\Sigma_Y} - \boldsymbol{\Sigma_{YX}}\boldsymbol{\Sigma_X^{-1}}\boldsymbol{\Sigma_{XY}}|} \\ =& \frac{1}{2}\log\frac{1}{|I+Q|^2} \\ =& -\log|I+Q| \end{aligned} \tag{13}$$

Since we want the mutual information to be greater than 0, we can formulate Equation 13 as an optimization problem:

$$\begin{aligned} &\max_Q MI(\mathcal{T},Q\boldsymbol{X}) \\ &s.t.\ rank(I+Q) = k \\ &\quad Q \sim I \\ &\quad [I+Q]_{ii} \geq [I+Q]_{ij}, i \neq j \\ &\quad \|[I+Q]_i\|_1 = 1 \end{aligned} \tag{14}$$

where $\sim$ means the row equivalence. The key to determining whether the linear transform is positive noise or not lies in the matrix of Q. The most important step is to ensure that $I + Q$ is reversible, which is $|(I+Q)| \neq 0$. The third constraint is to make the trained classifier get enough information about a specific image and correctly predict the corresponding label. For example, for an image X_1 perturbed by another image X_2, the classifier obtained dominant information from X_1 so that it can predict the label Y_1. However, if the perturbed image X_2 is dominant, the classifier can hardly predict the correct label Y_1 and is more likely to predict as Y_2. The fourth constraint is to maintain the norm of latent representations. More in-depth discussion and linear transform noise added to the image level are provided in the supplementary.

Salt-and-pepper Noise The salt-and-pepper noise is a common multiplicative noise for images. The image can exhibit unnatural changes, such as black pixels in bright areas or white pixels in dark areas, specifically as a result of the signal disruption caused by sudden strong interference or bit transmission errors. In the Salt-and-pepper noise case, the mutual information is:

$$\begin{aligned}
MI(\mathcal{T},\boldsymbol{\epsilon}) \overset{\star}{=} & H(\boldsymbol{Y};\boldsymbol{X}) - H(\boldsymbol{X}) - (H(\boldsymbol{Y};\boldsymbol{X}\boldsymbol{\epsilon}) - H(\boldsymbol{X})) \\
= & H(\boldsymbol{Y};\boldsymbol{X}) - H(\boldsymbol{Y};\boldsymbol{X}\boldsymbol{\epsilon}) \\
= & -\sum_{\boldsymbol{X}\in\mathcal{X}}\sum_{\boldsymbol{Y}\in\mathcal{Y}} p(\boldsymbol{X},\boldsymbol{Y})\log p(\boldsymbol{X},\boldsymbol{Y}) - \sum_{\boldsymbol{X}\in\mathcal{X}}\sum_{\boldsymbol{Y}\in\mathcal{Y}}\sum_{\boldsymbol{\epsilon}\in\mathcal{E}} p(\boldsymbol{X}\boldsymbol{\epsilon},\boldsymbol{Y})\log p(\boldsymbol{X}\boldsymbol{\epsilon},\boldsymbol{Y}) \\
= & \mathbb{E}\left[\log\frac{1}{p(\boldsymbol{X},\boldsymbol{Y})}\right] - \mathbb{E}\left[\log\frac{1}{p(\boldsymbol{X}\boldsymbol{\epsilon},\boldsymbol{Y})}\right] \\
= & \mathbb{E}\left[\log\frac{1}{p(\boldsymbol{X},\boldsymbol{Y})}\right] - \mathbb{E}\left[\log\frac{1}{p(\boldsymbol{X},\boldsymbol{Y})}\right] - \mathbb{E}\left[\log\frac{1}{p(\boldsymbol{\epsilon})}\right] \\
= & -H(\epsilon)
\end{aligned} \tag{15}$$

Obviously, the mutual information is smaller than 0, which indicates the complexity is increasing when injecting salt-and-pepper noise into the deep model. As can be foreseen, the salt-and-pepper noise is pure detrimental noise. More details and Salt-and-pepper added to the image level are in the supplementary.

4 Experiments

In this section, we conduct extensive experiments to explore the influence of various types of noises on deep learning models. We employ popular deep learning architectures, including both CNNs and ViTs, and show that the two kinds of deep models can benefit from the positive noise. We employ deep learning models of various scales, including ViT-Tiny (ViT-T), ViT-Small (ViT-S), ViT-Base (ViT-B), and ViT-Large (ViT-L) for Vision Transformers (ViTs), and ResNet-18, ResNet-34, ResNet-50, and ResNet-101 for ResNet architecture. The details of deep models are presented in the supplementary. Without specific instructions, the noise is injected at the last layer of the deep models. Note that for ResNet models, the number of macro layers is 4, and for each macro layer, different scale ResNet models have different micro sublayers. For example, for ResNet-18, the number of macro layers is 4, and for each macro layer, the number of micro sublayers is 2. The noise is injected at the last micro sublayer of the last macro layer for ResNet models. More experimental settings for ResNet and ViT are detailed in the supplementary.

4.1 Noise Setting

We utilize the standard normal distribution to generate Gaussian noise in our experiments, ensuring that the noise has zero mean and unit variance. Gaussian noise can be expressed as:

$$\epsilon \sim \mathcal{N}(0,1) \tag{16}$$

For linear transform noise, we use a quality matrix of as:

$$Q = -\alpha I + \alpha f(I) \tag{17}$$

where I is the identity matrix, α represents the linear transform strength and f is a row cyclic shift operation switching row to the next row, for example, in a 3×3 matrix, f will move Row 1 to Row 2, Row 2 to Row 3, and Row 3 to Row 1. For salt-and-pepper noise, we also use the parameter α to control the probability of the emergence of salt-and-pepper noise, which can be formulated as:

$$\begin{cases} max(X) & \text{if } p < \alpha/2 \\ min(X) & \text{if } p > 1-\alpha/2 \end{cases} \tag{18}$$

Table 1: ResNet with different kinds of noise on ImageNet. Vanilla means the vanilla model without noise. Accuracy is shown in percentage. Gaussian noise used here is subjected to standard normal distribution. Linear transform noise used in this table is designed to be positive noise. The difference is shown in the bracket.

Model	ResNet-18	ResNet-34	ResNet-50	ResNet-101
Vanilla	63.90 (+0.00)	66.80 (+0.00)	70.00 (+0.00)	70.66 (+0.00)
+ Gaussian Noise	62.35 (-1.55)	65.40 (-1.40)	69.62 (-0.33)	70.10 (-0.56)
+ Linear Transform Noise	**79.62 (+15.72)**	**80.05 (+13.25)**	**81.32 (+11.32)**	**81.91 (+11.25)**
+ Salt-and-pepper Noise	55.45 (-8.45)	63.36 (-3.44)	45.89 (-24.11)	52.96 (-17.70)

Table 2: ViT with different kinds of noise on ImageNet. Vanilla means the vanilla model without injecting noise. Accuracy is shown in percentage. Gaussian noise used here is subjected to standard normal distribution. Linear transform noise used in this table is designed to be positive noise. The difference is shown in the bracket. Note **ViT-L is overfitting on ImageNet** [13] [34].

Model	ViT-T	ViT-S	ViT-B	ViT-L
Vanilla	79.34 (+0.00)	81.88 (+0.00)	84.33 (+0.00)	88.64 (+0.00)
+ Gaussian Noise	79.10 (-0.24)	81.80 (-0.08)	83.41 (-0.92)	85.92 (-2.72)
+ Linear Transform Noise	**80.69 (+1.35)**	**87.27 (+5.39)**	**89.99 (+5.66)**	**88.72 (+0.08)**
+ Salt-and-pepper Noise	78.64 (-0.70)	81.75 (-0.13)	82.40 (-1.93)	85.15 (-3.49)

where p is a probability generated by a random seed, $\alpha \in [0, 1)$, and X is the representation of an image.

4.2 Image Classification Results

We implement extensive experiments on large-scale datasets such as ImageNet [11] and small-scale datasets such as TinyImageNet [17] using ResNets and ViTs.

4.2.1 CNN Family

The results of ResNets with different noises on ImageNet are in Table 1. As shown in the table, with the design of linear transform noise to be positive noise (PN), ResNet improves the classification accuracy by a large margin. While the salt-and-pepper, which is theoretically harmful noise (HN), degrades the models. Note we did not utilize data augmentation techniques for ResNet experiments except for normalization. The significant results show that positive noise can effectively improve classification accuracy by reducing task complexity.

4.2.2 ViT Family

The results of ViT with different noises on ImageNet are in Table 2. Since the ViT-L is overfitting on the ImageNet [13] [34], the positive noise did not work well on the ViT-L. As shown in the table, the existence of positive noise improves the classification accuracy of ViT by a large margin compared to vanilla ViT. The comparisons with previously published works, such as DeiT [38], SwinTransformer [20], DaViT [12], and MaxViT [39], are shown in Table 3, and our positive noise-empowered ViT achieved the new state-of-the-art result. Note that the JFT-300M and JFT-4B datasets are private and not publicly available [35], and we believe that ViT large and above will benefit from positive noise significantly if trained on larger JFT-300M or JFT-4B, which is theoretically supported in section 4.4.

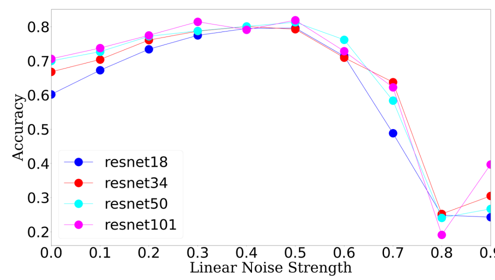
4.3 Ablation Study

We also proactively inject noise into variants of ViT, such as DeiT [38], Swin Transformer [20], BEiT [3], and ConViT [9], and the results show that positive noise could benefit various variants of ViT by improving classification accuracy significantly. The results of injecting noise to variants of ViT are reported in the supplementary. We also did ablation studies on the strength of linear transform noise and the injected layer. The results are shown in Fig. 2. We can observe that the deeper layer the positive noise injects, the better prediction performance the model can obtain. There are reasons behind this phenomenon. First, the latent features of input in the deeper layer have better

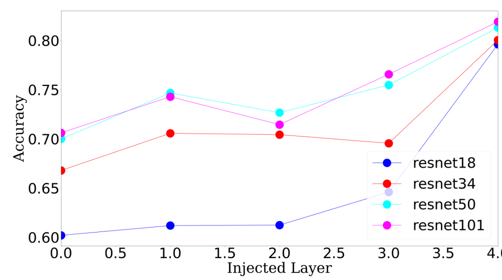
Table 3: Comparison between Positive Noise Empowered ViT with other ViT variants. Top 1 Accuracy is shown in percentage. Here PN is the positive noise, i.e., linear transform noise.

Model	Top1 Acc.	Params.	Image Res.	Pretrained Dataset
ViT-B [13]	84.33	86M	224 $\times$ 224	ImageNet 21k
DeiT-B [38]	85.70	86M	224 $\times$ 224	ImageNet 21k
SwinTransformer-B [20]	86.40	88M	384 $\times$ 384	ImageNet 21k
DaViT-B [12]	86.90	88M	384 $\times$ 384	ImageNet 21k
MaxViT-B [39]	88.82	119M	512 $\times$ 512	JFT-300M (Private)
ViT-22B [10]	89.51	21743M	224 $\times$ 224	JFT-4B (Private)
ViT-B+PN	**89.99**	86M	224 $\times$ 224	ImageNet 21k
ViT-B+PN	**91.37**	86M	384 $\times$ 384	ImageNet 21k

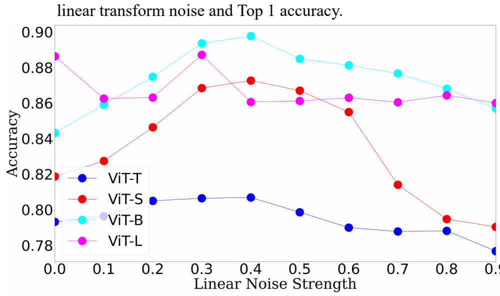
representations than those in shallow layers; second, injection to shallow layers obtain less mutual information gain because of trendy replacing Equation 8 with Equation 7. More results on the small dataset TinyImageNet can be found in the supplementary.



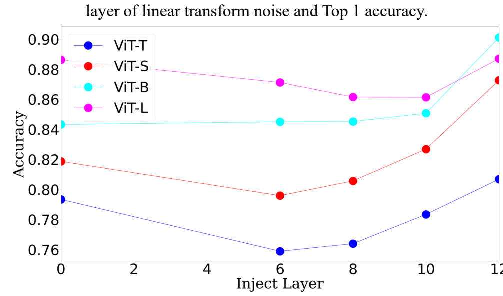
(a) The relationship of the CNN family between the strength of linear transform noise and Top 1 accuracy.



(b) The relationship of the CNN family between the injected layer of linear transform noise and Top 1 accuracy.



(c) The relationship of the ViT family between the strength of linear transform noise and Top 1 accuracy.



(d) The relationship of the ViT family between the injected layer of linear transform noise and Top 1 accuracy.

Figure 2: The relationship between the linear transform noise strength and the top 1 accuracy, and between the injected layer and top 1 accuracy. Parts (a) and (b) are the results of the CNN family, while parts (c) and (d) are the results of the ViT family. For parts (a) and (c) the linear transform noise is injected at the last layer. For parts (b) and (d), the influence of positive noise on different layers is shown. Layers 6, 8, 10, and 12 in the ViT family are chosen for the ablation study.

4.4 Optimal Quality Matrix

As shown in Equation 14, it is interesting to learn about the optimal quality matrix of Q that maximizes the mutual information while satisfying the constraints. This equals minimizing the determinant of the matrix sum of I and Q. Here, we directly give out the optimal quality matrix of Q as:

$$Q_{optimal} = \text{diag}\left(\frac{1}{k+1} - 1, \ldots, \frac{1}{k+1} - 1\right) + \frac{1}{k+1}\mathbf{1}_{k \times k} \tag{19}$$

where k is the number of data samples. And the corresponding upper boundary of the mutual information as:

$$MI(\mathcal{T}, Q_{optimal}\boldsymbol{X}) = (k-1)\log{(k+1)} \tag{20}$$

Table 4: Top 1 accuracy on ImageNet with the optimal quality matrix of linear transform noise.

Model	Top1 Acc.	Params.	Image Res.	Pretrained Dataset
ViT-B+Optimal Q	**93.87**	86M	224×224	ImageNet 21k
ViT-B+Optimal Q	**95.12**	86M	384×384	ImageNet 21k

Table 5: Comparison with various ViT-based methods on **Office-Home**.

Method	Ar2Cl	Ar2Pr	Ar2Re	Cl2Ar	Cl2Pr	Cl2Re	Pr2Ar	Pr2Cl	Pr2Re	Re2Ar	Re2Cl	Re2Pr	Avg.
ViT-B[13]	54.7	83.0	87.2	77.3	83.4	85.6	74.4	50.9	87.2	79.6	54.8	88.8	75.5
TVT-B[44]	74.9	86.8	89.5	82.8	88.0	88.3	79.8	71.9	90.1	85.5	74.6	90.6	83.6
CDTrans-B[43]	68.8	85.0	86.9	81.5	87.1	87.3	79.6	63.3	88.2	82.0	66.0	90.6	80.5
SSRT-B [36]	75.2	89.0	91.1	85.1	88.3	90.0	85.0	74.2	91.3	85.7	78.6	91.8	85.4
ViT-B+PN	**78.3**	**90.6**	**91.9**	**87.8**	**92.1**	**91.9**	**85.8**	**78.7**	**93.0**	**88.6**	**80.6**	**93.5**	**87.7**

The details are provided in the supplementary. We find that the upper boundary of the mutual information of injecting positive noise is determined by the number of data samples, i.e., the scale of the dataset. Therefore, the larger the dataset, the better effect of injecting positive noise into deep models. With the optimal quality matrix and the top 1 accuracy of ViT-B on ImageNet can be further improved to 95%, which is shown in Table 4.

Table 6: Comparison with various ViT-based methods on **Visda2017**.

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
ViT-B[13]	97.7	48.1	86.6	61.6	78.1	63.4	94.7	10.3	87.7	47.7	94.4	35.5	67.1
TVT-B[44]	92.9	85.6	77.5	60.5	93.6	98.2	89.4	76.4	93.6	92.0	91.7	55.7	83.9
CDTrans-B[43]	97.1	90.5	82.4	77.5	96.6	96.1	93.6	**88.6**	**97.9**	86.9	90.3	62.8	88.4
SSRT-B [36]	**98.9**	87.6	**89.1**	**84.8**	98.3	**98.7**	**96.3**	81.1	94.9	97.9	94.5	43.1	88.8
ViT-B+PN	98.8	**95.5**	84.8	73.7	**98.5**	97.2	95.1	76.5	95.9	**98.4**	**98.3**	**67.2**	**90.0**

4.5 Domain Adaption Results

Unsupervised domain adaptation (UDA) aims to learn transferable knowledge across the source and target domains with different distributions [25] [42]. Recently, transformer-based methods achieved SOTA results on UDA, therefore, we evaluate the ViT-B with the positive noise on widely used UDA benchmarks. Here the positive noise is the linear transform noise identical to that used in the classification task. The positive noise is injected into the last layer of the model, the same as the classification task. The datasets include **Office Home** [41] and **VisDA2017** [26]. Detailed datasets introduction and experiments training settings are in the supplementary. The objective function is borrowed from TVT [44], which is the first work that adopts Transformer-based architecture for UDA. The results are shown in Table 5 and 6. The ViT-B with positive noise achieves better performance than the existing works. These results show that positive noise can improve model generality, therefore, benefit deep models in domain adaptation tasks.

5 Conclusion

This study presents a comprehensive investigation into the influence of various common noise types on deep learning models, including Gaussian noise, linear transform noise, and salt-and-pepper noise. We demonstrate that, under certain conditions, linear transform noise can have a positive effect on deep models. Our experiments show that injecting the positive noise in latent space can significantly enhance the prediction performance of deep models on image classification tasks, leading to new state-of-the-art results on ImageNet. The findings of this study have a broad impact on future research and may contribute to the development of more accurate models and their improved performance in real-world applications. Moreover, we are excited to explore the potential of positive noise in more deep learning tasks.

References

[1] Osama K. Al-Shaykh and Russell M. Mersereau. Lossy compression of noisy images. *IEEE Transactions on Image Processing*, 7(12):1641–1652, 1998.

[2] Wissam A. Albukhanajer, Johann A. Briffa, and Yaochu Jin. Evolutionary multiobjective image feature extraction in the presence of noise. *IEEE Transactions on Cybernetics*, 45(9):1757–1768, 2014.

[3] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[4] Roberto Benzi, Alfonso Sutera, and Angelo Vulpiani. The mechanism of stochastic resonance. *Journal of Physics A: mathematical and general*, 14(11):L453, 1981.

[5] George EP Box and David R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–243, 1964.

[6] Sebastian Braun, Hannes Gamper, Chandan KA Reddy, and Ivan Tashev. Towards efficient models for real-time deep noise suppression. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 656–660, 2021.

[7] Raymond H. Chan, Chung-Wa Ho, and Mila Nikolova. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Transactions on image processing*, 14(10):1479–1485, 2005.

[8] Thomas M. Cover. Elements of information theory. *John Wiley & Sons*, 1999.

[9] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.

[10] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, and et al. Scaling vision transformers to 22 billion parameters. *arXiv preprint arXiv:2302.05442 (2023)*, 2023.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Feifei Li. Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009.

[12] Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jindong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *In Computer Vision–ECCV 2022: 17th European Conference*, pages 74–92, 2022.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *arXiv preprint arXiv:2010.11929*, 2020.

[14] Changyong Feng, Hongyue Wang, Naiji Lu, Tian Chen, Hua He, Ying Lu, and Xin M. Tu. Log-transformation and its implications for data analysis. *Shanghai archives of psychiatry*, 26(2):105, 2014.

[15] 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*, pages 770–778, 2016.

[16] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[17] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N 7*, (7), 2015.

[18] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

[19] Xuelong Li. Positive-incentive noise. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[21] Peter McClintock. Can noise actually boost brain power? *Physics World*, 15(7), 2002.

[22] Toshio Mori and Shoichi Kai. Noise-induced entrainment and stochastic resonance in human brain waves. *Physical review letters*, 88(21), 2002.

[23] Rich Ormiston, Tri Nguyen, Michael Coughlin, Rana X. Adhikari, and Erik Katsavounidis. Noise reduction in gravitational-wave data via deep learning. *Physical Review Research*, 2(3):033066, 2020.

[24] J. S. Owotogbe, T. S. Ibiyemi, and B. A. Adu. A comprehensive review on various types of noise in image processing. *int. J. Sci. eng. res*, 10(10):388–393, 2019.

[25] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[26] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

[27] Lis Kanashiro Pereira, Yuki Taya, and Ichiro Kobayashi. Multi-layer random perturbation training for improving model generalization efficiently. *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2021.

[28] Kamiar Radnosrati, Gustaf Hendeby, and Fredrik Gustafsson. Crackling noise. *IEEE Transactions on Signal Processing*, 68:3590–3602, 2020.

[29] Fabrizio Russo. A method for estimation and filtering of gaussian noise in images. *IEEE Transactions on Instrumentation and Measurement*, 52(4):1148–1154, 2003.

[30] James P. Sethna, Karin A. Dahmen, and Christopher R. Myers. Crackling noise. *Nature*, 410(6825):242–250, 2001.

[31] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, Cambridge, 2014.

[32] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[33] Jan Sijbers, Paul Scheunders, Noel Bonnet, Dirk Van Dyck, and Erik Raman. Quantification and improvement of the signal-to-noise ratio in a magnetic resonance image acquisition procedure. *Magnetic resonance imaging*, 14(10):1157–1163, 1996.

[34] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. In *arXiv preprint arXiv:2106.10270*, 2021.

[35] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *In Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[36] Tao Sun, Cheng Lu, Tianshuo Zhang, and Harbin Ling. Safe self-refinement for transformer-based domain adaptation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7191–7200, 2022.

[37] Sunil Thulasidasan, Tanmoy Bhattacharya, Jeff Bilmes, Gopinath Chennupati, and Jamal Mohd-Yusof. Combating label noise in deep learning using abstention. In *arXiv preprint arXiv:1905.10964*, 2019.

[38] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357, 2021.

[39] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *In Computer Vision–ECCV 2022: 17th European Conference*, pages 459–479, 2022.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017.

[41] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. *CVPR*, pages 5018–5027, 2017.

[42] Ying Wei, Yu Zhang, Junzhou Huang, and Qiang Yang. Transfer learning via learning to transfer. *ICML*, pages 5085–5094, 2018.

[43] Tongkun Xu, Weihua Chen, Fan Wang, Hao Li, and Rong Jin. Cdtrans: Cross-domain transformer for unsupervised domain adaptation. *ICLR*, pages 520–530, 2022.

[44] Jinyu Yang, Jingjing Liu, Ning Xu, and Junzhou Huang. Tvt: Transferable vision transformer for unsupervised domain adaptation. *WACV*, pages 520–530, 2023.

[45] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] Yes
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? **[TODO]** Yes
   (b) Did you describe the limitations of your work? **[TODO]** Yes
   (c) Did you discuss any potential negative societal impacts of your work? **[TODO]** No
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? **[TODO]** Yes
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? **[TODO]** Yes
   (b) Did you include complete proofs of all theoretical results? **[TODO]** Yes
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? **[TODO]** Yes

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **[TODO]** Yes

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? **[TODO]** No

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? No **[TODO]**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

    (a) If your work uses existing assets, did you cite the creators? **[TODO]** N/A

    (b) Did you mention the license of the assets? **[TODO]** N/A

    (c) Did you include any new assets either in the supplemental material or as a URL? **[TODO]** N/A

    (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? **[TODO]** N/A

    (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **[TODO]** N/A

5. If you used crowdsourcing or conducted research with human subjects...

    (a) Did you include the full text of instructions given to participants and screenshots, if applicable? **[TODO]** N/A

    (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? **[TODO]** N/A

    (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? **[TODO]** N/A