AdPE: ADVERSARIAL POSITIONAL EMBEDDINGS FOR PRETRAINING VISION TRANSFORMERS VIA MAE+

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised learning of vision transformers seeks to pretrain an encoder via pretext tasks without labels. Among them is the Masked Image Modeling (MIM) aligned with pretraining of language transformers by predicting masked patches as a pretext task. A criterion in unsupervised pretraining is the pretext task needs to be sufficiently hard to prevent the transformer encoder from learning trivial low-level features not generalizable well to downstream tasks. For this purpose, we propose an Adversarial Positional Embedding (AdPE) approach – It distorts the local visual structures by perturbing the position encodings so that the learned transformer cannot simply use the locally correlated patches to predict the missing ones. We hypothesize that it forces the transformer encoder to learn more discriminative features in a global context with stronger generalizability to downstream tasks. We will consider both absolute and relative positional encodings, where adversarial positions can be imposed both in the embedding mode and the coordinate mode. We will also present a new MAE+ baseline that brings the performance of the MIM pretraining to a new level with the AdPE. The experiments demonstrate that our approach can improve the fine-tuning accuracy of MAE by 0.8% and 0.4% over 1600 epochs of pretraining ViT-B and ViT-L on Imagenet1K. For the transfer learning task, it outperforms the MAE with the ViT-B backbone by 2.6% in mIoU on ADE20K, and by 3.2% in AP^{bbox} and 1.6% in AP^{mask} on COCO, respectively. These results are obtained with the AdPE being a pure MIM approach that does not use any extra models or external datasets for pretraining.

1 INTRODUCTION

Pretraining vision transformers (Dosovitskiy et al., 2020) effectively has received many attentions due to the potential of unifying transformer architectures across modalities. Among them is the Masked Image Modeling (MIM) (Bao et al., 2021)(He et al., 2022) that inherits the same idea in Bert pretraining of language transformers (Devlin et al., 2018). These MIM approaches seek to predict the masked patches as a pretext task to pretrain vision transformers.

A critical principle in unsupervised pretraining of deep networks is the pretext task ought to be sufficiently hard to avoid trivial solutions only focusing on low-level features to bypass the task (Robinson et al., 2021). For this purpose, adversarial pretraining of the CNNs has demonstrated tremendous successes in the context of contrastive learning (Hu et al., 2021; Kim et al., 2020; Robinson et al., 2021). For example, Hu et al. (2021) learn to generate hard negatives, forcing the network encoder to learn more discriminative features to distinguish hard negatives from their positive counterparts. By adopting harder pretext tasks (Robinson et al., 2021), the results showcase the adversarial pretraining is able to prevent the deep network from learning trivial low-level features with poor generalizability to downstream tasks.

Along this line of research, we aspire to develop an adversarial approach to effectively pretrain transformers in a MIM manner. While the adversarial contrastive learning seeks to learn hard negatives (Hu et al., 2021; Kalantidis et al., 2020; Robinson et al., 2020), no such adversaries exist in the MIM-based pretraining. Thus, we first need to answer the question of what to choose as the adversary in the MIM-pretrained transformer. In this paper, we propose that perturbing the positional encodings is a natural choice to adversarially pretrain the vision transformer. It spatially distorts the local visual structures through perturbed positional encodings, and thus prevents the transformer from learning trivial features by exploiting the strong correlations between masked patches and their local unmasked peers.

In this way, we hypothesize the transformer is adversarially pretrained to focus on global contexts that are more useful for the downstream tasks to predict masked patches. We will consider adversarial perturbations applied in the positional embedding space additively, or on the image coordinates through a differentiable positional embedding/indexing function. Both absolute (Shaw et al., 2018) and relative positional embeddings (Wu et al., 2021) will be considered for adversarial pretraining of transformers.

In addition, we will present a new MAE+ baseline by seamlessly fitting multi-crop tokenization to MAE (He et al., 2022). While multi-crop augmentation has been successfully applied in the contrastive pretraining of both CNNs (Caron et al., 2020) and transformers (Caron et al., 2021), we will demonstrate how this simple mechanism can be delicately designed with a lighter-weighted decoder having fewer input tokens than the existing MIM baseline (He et al., 2022). It allows to strike a better trade-off between the lighter decoder and more crops of tokenization to improve the MIM-pretraining. We will demonstrate this results in a more efficient MAE+ baseline reaching superior performances alongside the proposed Adversarial Positional Embeddings (AdPE).

The remainder of this paper is organized as follows. We will review the related works in Section 2. Then we will revisit the key idea behind the MIM and present a new MAE+ baseline in Section 3. We will elaborate on the proposed Adversarial Positional Embeddings (AdPE) in Section 4. Experiment results will be reported in Section 5, and we will conclude the paper in Section 6.

2 Related Works

We will review the works that are closely related with the proposed method from three aspects – MIM-based pretraining of transformers, absolute and relative positional encodings, and adversarial pretraining of deep networks.

2.1 Masked Image Modeling and Pretraining Vision Transformers

While it is natural to extend the contrastive learning approaches that achieve tremendous successes in pretraining the CNNs to pretrain the Vision Transformers (ViTs), Masked Image Modeling (MIM) (Bao et al., 2021)(He et al., 2022) provides an alternative way inspired by the success in the NLP domain (Devlin et al., 2018). This is expected to unify the transformer pretraining in computer vision, NLP and multmodality domains. The idea is simple – it aims to unsupervisedly train a transformer encoder by masking out some tokens and reconstructing them at the decoder end. Such a pretext task aims to learn a useful transformer backbone as its encoder to learn representations that are useful to predict the masked contents.

It is demonstrated that by masking out a large portion of input images (e.g., 75% masked out (He et al., 2022)), a hard MIM task is formulated that forces the encoder to learn useful clues in the long-range contexts to predict the missing patches. Masked tokens will be added either before (Bao et al., 2021) or after (He et al., 2022) the encoder. By only feeding the unmasked tokens through the encoder, the Masked Auto-Encoder (MAE) is able to reduce the computing costs for its encoder during the pretraining (He et al., 2022). However, it still needs to feed the masked tokens through the decoder to predict the missing patches.

2.2 Positional Encodings

The self-attention mechanism is unaware of position information by itself, and Positional Encodings (PEs) are thus required to represent the knowledge of where a token is in a sentence or in an image. Sinusoid embedding of positions was first proposed in the seminal paper (Vaswani et al., 2017). It transforms a hard-coded position into a Fourier basis through sine/cosine functions at different frequencies. Such an Absolute PE (APE) is variant when the sequence is shifted.

Alternatively, Relative PEs (RPEs) (Shaw et al., 2018; Dai et al., 2019; Ramachandran et al., 2019) aim to encode the positions between tokens based on their relative relations, such as their relative distances and positions (Wu et al., 2021). The resultant positional encodings are invariant when the sequence or the image is shifted. Various RPEs have been proposed. We will revisit them in Section 4.2.1 before discussing how adversaries can be imposed on the RPEs.

2.3 Adversarial Pretraining

While adversarial *training* has been intensively studied through adversarial examples (Szegedy et al., 2013)(Goodfellow et al., 2014)(Madry et al., 2017), adversarial *pretraining* of deep networks, especially CNNs, is receiving lots of attentions recently in the context of contrastive learning (Kim et al., 2020; Robinson et al., 2021). One of representative works in this line of research is to treat negative samples as adversarially learnable by maximizing instead of minimizing the InfoNCE loss (Hu et al., 2021). Hard negatives are thus generated directly, where they are continuously being pushed towards their positive counterparts so that more discriminative features must be learned to distinguish between positives and negatives for a query. The idea was also extended to incorporate learnable positives in a cooperative-adversarial fashion together with learnable negatives (Wang et al., 2022). All these methods focus on the adversarial pretraining for contrastive learning.

In this paper, we seek to adversarially pretrain vision transformers in the MIM fashion. In contrast to obtaining hard negatives in (Hu et al., 2021; Kalantidis et al., 2020; Robinson et al., 2020), we will impose adversarial perturbations on positional embeddings to distort the local visual structures spatially so that the pretrained transformers are forced to explore high-level features in long-range contexts to predict missing patches, instead of simply cheating on the strongly correlated patches in local neighborhood.

3 Masked Image Modeling and A New MAE+ Baseline

In this section, first we will briefly revisit the Masked Image Modeling (MIM)-based approaches for pretraining vision transformers. Then we will present a new MIM-pretraining baseline, **MAE+** based on multi-crop tokenization that seamlessly fits the state-of-the-art MAE (He et al., 2022), boosting performances with less computing cost. The MAE+ will be used as the new baseline to showcase the greater potential of the MIM-pretraining.

3.1 Masked Image Modeling and Masked Auto-Encoders

The Masked Image Modeling (MIM)-based methods pretrain vision transformers through an encoder-decoder architecture. Given an image, it is flattened to a sequence of tokens corresponding to a group of non-overlapping patches. A large portion of patch tokens in the input sequence will be masked out, being replaced with a mask token before feeding into the encoder (Bao et al., 2021). Alternatively, only unmasked patch tokens will be input to the encoder (He et al., 2022), and a shared learnable mask token will be used to represent each masked patch before feeding all tokens into the decoder to predict the missing patches.

The decoder can be composed of several layers of transformers (He et al., 2022), or simply consist of a simple fully-connected layer (Xie et al., 2022). The encoder is a backbone transformer that we wish to pretrain and use later in the downstream tasks. A mean-squared reconstruction error over masked patches is minimized to pre-train both the encoder and the decoder end-to-end through back-propagation in the Masked Auto-Encoder (MAE) architecture (He et al., 2022).

3.2 MAE+: A New Baseline for MIM-Pretraining of Vision Transformers

Typically, in a MIM-based approach (Bao et al., 2021; He et al., 2022), an input image of resolution 224×224 is divided into 14×14 tokens, each of which correspondences to a non-overlapping patch of 16×16. In the MAE (He et al., 2022), 75% tokens are masked out, leaving only 49 unmasked tokens that will be fed through an ViT encoder. After that, a group of learnable masked tokens will be concatenated with the unmasked tokens output from the encoder, forming a total of 196 tokens which will then go through several layers of transformer decoders to reconstruct the patches of masked tokens.

Such a MAE architecture has some drawbacks preventing it from releasing its full potentials. First, although only a much smaller number of 49 unmasked tokens are fed into the encoder, the decoder still has a full number of 196 tokens as input. The computational complexity squared in the number of the decoder tokens still incurs heavy costs to pretrain a ViT network. Second, Although multi-crop augmentation has been studied in both CNNs (Caron et al., 2020) and vision transformers (Caron et al., 2021) for contrastive learning, a delicate mechanism tailored to the masked image modeling has yet to be developed considering both efficiency and accuracy.

To this end, we propose a new MIM baseline named MAE+, and show that it can seamlessly fit the MIM-pretraining of transformers. Particularly, we randomly crop a full-sized image by half to a small scale of 112×112, but still tokenize it with non-overlapping patches of 16×16. This will result in exactly $49 (= 7 \times 7)$ tokens, the same number of unmasked tokens fed into the MAE encoder. Then we randomly mask out 75% tokens. Unlike the MAE, we allow all these 49 tokens to feed through the encoder, no matter if they are masked or not, which does not incur more computational burden than the MAE encoder. Now, the decoder will only take these 49 tokens as input. In contrast, the MAE decoder has a total of 196 masked and unmasked tokens as its input on a full-sized image, whose computing complexity is up to 16 times that of a cropped one.

The saved computational cost in the decoder allows us to have multiple crops without increasing the computing overhead. One can even adopt a simple fully-connected layer as the decoder as in the SimMIM (Xie et al., 2022). More crops of tokenization can yield better performances, and a delicate trade-off can be reached between the lower complexity of the decoder and more tokenized crops per image to pretrain a vision transformer to its great potential. Once pretrained, the transformer encoder is used conventionally on full-sized images as a backbone in downstream tasks.

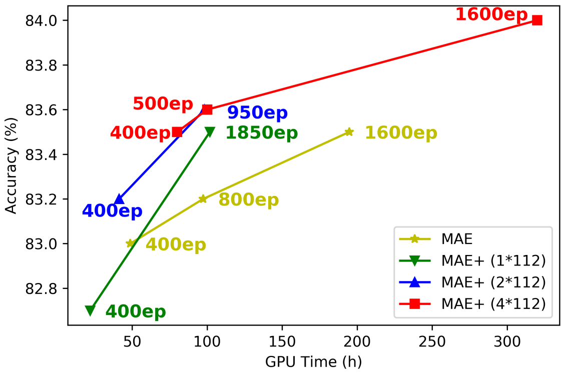


Figure 1: The figure illustrates the top-1 accuracy on Imagenet1K achieved by MAE and MAE+ with the ViT-B backbone. It compares the accuracy against the GPU hours (on a single GPU server with eight V100 Nvidia cards) used for network pretraining. The $k * 112$ in the parenthesis denotes k of $112 * 112$ sub-images are cropped to tokenize an input image. Note that the pretrained backbone is fine-tuned on the original $224 * 224$ images without multi-crop tokenization following the evaluation protocol (He et al., 2022).

Figure 1 compares the top-1 accuracy achieved by MAE and MAE+ with the ViT-B backbone by pretraining on Imagenet1k dataset. The same encoder and decoder transformers are adapted except the MAE+ uses the multi-crop tokenization. The same evaluation protocol as in MAE (He et al., 2022) is adopted by fine-tuning the pretrained encoder with Imagenet1K labels on the original $224 * 224$ images *without* multi-crop tokenization. The results show that MAE+ can remarkably improve the accuracy while reducing the computing cost. For example, with half of GPU hours, even a single crop MAE+ (1*112) can achieve the same top-1 accuracy as the MAE pretrained for 1600 epochs. With more crops k, more pretraining cost can be saved with even higher accuracy. The details about the experiment setting on the MAE+ are discussed in Appendix A. In the following, we will adopt MAE+ with four crops $(4 * 112)$ as the baseline model for the proposed approach.

4 Adversarial Positional Embeddings

In this section, we will present the proposed Adversarial Positional Embeddings (AdPE). On a high level, it seeks to make it harder to reconstruct the masked tokens. In particular, the AdPE will distort the local spatial structures, which will prevent the MIM from merely leveraging the local correlations between tokens to reconstruct those masked ones. In this way, the learned representation can capture more useful high-level features in global contexts.

Adversarial perturbations will be added onto the positional embeddings in two ways: in the embedding space (i.e., embedding mode) and in the spatial coordinates (i.e., coordinate mode). We will consider both absolute positional embeddings and relative positional embeddings for the AdPE.

4.1 Adversarial Absolute Positional Embeddings

Consider a token representation $\boldsymbol{t}_i \in \mathbb{R}^d$ at an image location (x_i, y_i). Before it is fed through a transformer, a positional encoding $\boldsymbol{p}_{x_i,y_i}$ of its coordinates is added onto $\boldsymbol{t}_i$ to give rise to a resultant position-aware token.

We seek to distort the local structures represented by the positional embeddings, so that the MIM cannot merely explore the patch-level correlations in a local affinity to predict the masked tokens. For this, there are two different ways to add adversarial perturbations to the positional information.

4.1.1 Embedding Mode Adversaries

The most straight way is to directly add an adversarial perturbation $\boldsymbol{a} \in \mathbb{R}^d$ onto the positional embedding $\boldsymbol{p}_{x_i,y_i}$. This results in a perturbed positional embedding $\boldsymbol{p}_{x_i,y_i} + \boldsymbol{a}$ being fed into a transformer layer. Leaving all the trainable network weights in $\boldsymbol{\theta}$, the MIM objective becomes

$$\min_{\boldsymbol{\theta}} \max_{\|\boldsymbol{a}\|_q \le \epsilon} \mathcal{L}(\{\boldsymbol{p}_{x_i,y_i} + \boldsymbol{a} | i \in \mathbb{M}\}; \boldsymbol{\theta})$$

where $\mathcal{L}$ is the MAE loss that minimizes the mean squared reconstruction errors over masked tokens, and $\|\boldsymbol{a}\|_q \le \epsilon$ is the ℓ_q-norm constraint on the magnitude of the perturbation. The adversarial perturbations are only added to the set $\mathbb{M}$ of masked tokens, and better performances have been observed by sharing adversaries $\boldsymbol{a}$ over masked tokens. To solve this constrained minimax problem, we will adopt the following two strategies throughout the paper.

Parallel mode updates. We adopt the stochastic gradient to update $\boldsymbol{\theta}$ and $\boldsymbol{a}$. However, unlike the other adversarial approach (Szegedy et al., 2013) that sequentially update two adversarial players, these two parts are *simultaneously* updated via negative gradient (i.e., minimizing the loss over $\boldsymbol{\theta}$) and positive gradient (i.e., maximizing the loss over $\boldsymbol{a}$). In other words, for each iteration, we only feed a sample forward and backward *once* through the network to update $\boldsymbol{\theta}$ and $\boldsymbol{a}$ in parallel, without alternating between them [1]. We do not find any significant difference between sequential and parallel updates in performance but the latter can save up to half of the computing time per iteration.

Projected gradient descent. The ℓ_q-norm constraint can be enforced by the Projected Gradient Descent (PGD) iteratively [2]. Once $\boldsymbol{a}$ is updated each time, it is immediately projected onto the ℓ_q-ball with a radius of ϵ to meet the constraint, that is

$$\boldsymbol{a} \leftarrow \Pi_{\mathbb{S}}(\boldsymbol{a} + \alpha \nabla_{\boldsymbol{a}} \mathcal{L}(\{\boldsymbol{p}_{x_i,y_i} + \boldsymbol{a} | i \in \mathbb{M}\}; \boldsymbol{\theta}))$$

with a learning rate α, and the constraint set $\mathbb{S} = \{\boldsymbol{a} | \|\boldsymbol{a}\|_q \le \epsilon\}$. For the ℓ_2 norm, the projection clips the resultant perturbation to a maximum length of ϵ, i.e., $\Pi_{\mathbb{S}}(\boldsymbol{v}) = \frac{\min(\epsilon, \|\boldsymbol{v}\|_2) \cdot \boldsymbol{v}}{\|\boldsymbol{v}\|_2}$. For the ℓ_∞ norm, the projection clips the resultant perturbation element-wise to a maximum absolute value of ϵ, i.e., $[\Pi_{\mathbb{S}}(\boldsymbol{v})]_l = \min(|v_l|, \epsilon) \cdot \operatorname{sign}(v_l)$.

We will adopt both strategies to update the adversarial perturbation unless stated otherwise. In experiments, the computing overhead for the PGD-based AdPE with the parallel mode updates is less than 1.5% compared to the MAE+ baseline.

4.1.2 Coordinate Mode Adversaries

Adding adversaries in the positional embedding space is implicit in how the underly spatial structure is perturbed. A more direct way is to add the adversaries $\boldsymbol{\delta} = (\delta_x, \delta_y)$ to the underlying coordinates,

[1] In sequential mode, each time after updating and fixing one part of parameters, one needs to feed forward and backward an image again before the other part being updated and fixed.

[2] Alternatively, linearizing the loss $\mathcal{L}$ can lead to a direct update to $\boldsymbol{a}$ (Goodfellow et al., 2014)(Kurakin et al., 2016). For the ℓ_2 norm, we have $\boldsymbol{a} = \frac{\epsilon \boldsymbol{g}}{\|\boldsymbol{g}\|_2}$, and for the ℓ_∞ norm, $\boldsymbol{a} = \epsilon \operatorname{sign}(\boldsymbol{g})$, where $\boldsymbol{g} = \nabla_{\boldsymbol{a}} \mathcal{L}(\{\boldsymbol{p}_{x_i,y_i} + \boldsymbol{a}\}_i; \boldsymbol{\theta})|_{\boldsymbol{a}=\boldsymbol{0}}$ is the loss gradient at zero perturbation. However, the iterative update via PGD can find more sophisticated adversarial perturbations than the direct approach (Tramèr et al., 2017).

resulting in disturbed coordinates $(x_i + \delta_x, y_i + \delta_y)$ and the corresponding positional embedding $\boldsymbol{p}_{x_i+\delta_x, y_i+\delta_y}$.

In this case, the objective of the MIM task becomes

$$\min_{\boldsymbol{\theta}} \max_{\|\boldsymbol{\delta}\|_q \leq \epsilon} \mathcal{L}(\{\boldsymbol{p}_{x_i+\delta_x, y_i+\delta_y} | i \in \mathbb{M}\}; \boldsymbol{\theta})$$

such that the adversaries are maximized to distort the spatial coordinates. Usually the absolute positional embedding is a sinusoid function (Vaswani et al., 2017),

$$\boldsymbol{p}_{x_i+\delta_x, y_i+\delta_y} = \begin{bmatrix} \sin((x_i+\delta_x)/10000^{2j/d_{\text{model}}}) \\ \cos((x_i+\delta_x)/10000^{2j/d_{\text{model}}}) \end{bmatrix}_j \oplus \begin{bmatrix} \sin((y_i+\delta_y)/10000^{2j/d_{\text{model}}}) \\ \cos((y_i+\delta_y)/10000^{2j/d_{\text{model}}}) \end{bmatrix}_j$$

which is differential in $\boldsymbol{\delta}$ so it can be learned via back-propagation. Here, d_{model} is the dimension compatible with the model, and $\oplus$ denotes the vector concatenation.

4.2 Adversarial Relative Positional Embeddings

Like in absolute positional embeddings, the relative positional embeddings also have two adversarial modes on embeddings and coordinates, respectively.

4.2.1 Relative Positional Encodings

Let us first revisit the relative positional encodings (Wu et al., 2021). Consider a pair of token representations $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$. We have a scaled correlation matrix $\boldsymbol{e}$ whose entries e_{ij} is computed as

$$e_{ij} = \frac{(\boldsymbol{t}_i \boldsymbol{W}^Q \cdot \boldsymbol{t}_j \boldsymbol{W}^K) + b_{ij}}{\sqrt{d_z}}$$

where the bias is either $b_{ij} = r_{ij}$ or $b_{ij} = \boldsymbol{t}_i \boldsymbol{W}^Q \cdot \boldsymbol{r}_{ij}$ with $\boldsymbol{r}_{ij} \in \mathbb{R}^{d_z}$.

Here $\boldsymbol{r}_{ij}$ is the relative positional encoding for a token pair $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$, which is a learnable vector. A softmax function is applied to transform e_{ij} into an attention matrix $\boldsymbol{\alpha}$, and the output token can be written as

$$\boldsymbol{z}_i = \sum_{j=1}^{n} \alpha_{ij} (\boldsymbol{t}_j \boldsymbol{W}^V + \boldsymbol{r}_{ij}^V)$$

with $\boldsymbol{r}_{ij}^V \in \mathbb{R}^{d_z}$ is a value embedding of relative positions between the two tokens. For simplicity, we can use a unified representation $\boldsymbol{r}_{ij}$ to denote the positional embeddings in different cases.

An index function $\text{Ind} : \mathbb{I} \times \mathbb{I} \rightarrow \mathbb{I}$ is defined to map a pair of tokens (i, j) to an integer index $\text{Ind}(i, j)$ such that a learnable positional embedding $\boldsymbol{p}_{\text{Ind}(i,j)}$ can be retrieved from a dictionary. For an image, its x-coordinate and y-coordinate will be mapped separately to two indices denoted by $\text{Ind}_x(i, j)$ and $\text{Ind}_y(i, j)$. Then the relative positional encoding $\boldsymbol{r}_{ij}$ of the two tokens is given by

$$\boldsymbol{r}_{ij} \triangleq \boldsymbol{p}_{[\text{Ind}_\text{x}(i,j), \text{Ind}_\text{y}(i,j)]}$$

with such a 2D index $[\text{Ind}_\text{x}(i, j), \text{Ind}_\text{y}(i, j)]$ in square brackets to a dictionary of relative positional embeddings $\mathbb{P} = \{\boldsymbol{p}_{[k,l]} | k, l = -\beta, \cdots, +\beta\}$ (see details below).

4.2.2 Embedding Mode Adversaries

It is straight to add adversarial perturbation to the relative positional encodings directly,

$$\boldsymbol{p}_{[\text{Ind}_\text{x}(i,j), \text{Ind}_\text{y}(i,j)]} + \boldsymbol{a},$$

where $\boldsymbol{a}$ is such an additive adversarial perturbation in the embedding mode as in the aforementioned APE case. Now the MIM task becomes

$$\min_{\boldsymbol{\theta}, \boldsymbol{p}} \max_{\|\boldsymbol{a}\|_q \leq \epsilon} \mathcal{L}(\{\boldsymbol{p}_{[\text{Ind}_\text{x}(i,j), \text{Ind}_\text{y}(i,j)]} + \boldsymbol{a} | i \in \mathbb{M}\}; \boldsymbol{\theta})$$

where the constrained adversarial perturbation $\boldsymbol{a}$ is applied so long as the first token is masked in a pairwise relation no matter if the second one is masked or not. While the positional embeddings $\boldsymbol{p}$'s are learned by minimizing the MAE loss as the other weights $\boldsymbol{\theta}$, the perturbation is learned in an adversarial manner by maximizing the loss. Relative positional encodings are applied to various layers of transformers. For each layer of transformer, a distinct adversarial perturbation is applied and learned. This allows us to perturb positional structures between tokens differently to model the transformer representation on various scales.

4.2.3 Coordinate Mode Adversaries

The adversarial perturbations can also be added to the coordinates (x_i, y_i) and (x_j, y_j) directly. For example, a piece-wise function $g : \mathbb{R} \to \mathbb{I}$ in (Wu et al., 2021) has been introduced to define the index function such that $\mathrm{Ind_x}(i, j) = g(x_i - x_j)$ and $\mathrm{Ind_y}(i, j) = g(y_i - y_j)$, which is able to distribute various ranges of attention by the relative distance between two tokens i and j as illustrated in Figure 2. Then the adversarial perturbations δ_x and δ_y on the relative coordinates in these two index functions give rise to an adversarial positional embedding $\boldsymbol{p}_{[g(x_i - x_j + \delta_x), g(y_i - y_j + \delta_y)]}$.

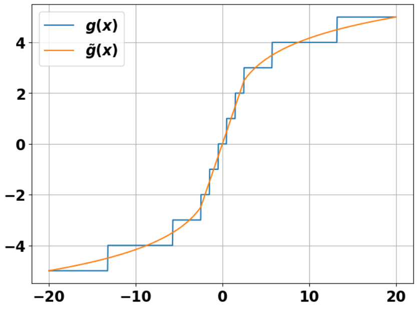


Figure 2: Illustration of the integer-valued index function g and its relaxed form $\tilde{g}$.

However, the gradient of the adversarial positional embedding $\boldsymbol{p}_{[g(x_i - x_j + \delta_x), g(y_i - y_j + \delta_y)]}$ over the perturbation $\boldsymbol{\delta}$ is ill-defined for back-propagation for the step-wise g. A small perturbation on the image coordinates either incurs an abrupt change in indexing g to another positional embedding, resulting in an infinitely large gradient; or makes no change at all, leading to a vanishing gradient.

To mitigate this problem, we relax the integer index g to a real-valued function $\widetilde{g}$. For example, by removing the round operation, the piece-wise linear function is relaxed to

$$\widetilde{g}(x) = \begin{cases} x & |x| \le \alpha \\ \mathrm{sign}(x) \cdot \min(\beta, \alpha + \dfrac{\ln(|x|/\alpha)}{\ln(\gamma/\alpha)}(\beta - \alpha)) & |x| > \alpha \end{cases}$$

Figure 2 compares the integer index g with its relaxed form $\widetilde{g}$.

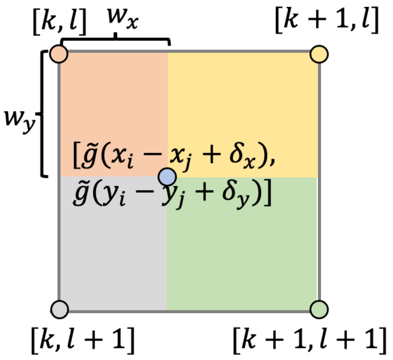

$[k, l]$
w_x
$[k + 1, l]$
w_y
$[\tilde{g}(x_i - x_j + \delta_x),$
$\tilde{g}(y_i - y_j + \delta_y)]$
$[k, l + 1]$
$[k + 1, l + 1]$

Figure 3: The bilinear interpolation of four nearest positional embeddings via $\tilde{g}$.

To have differentiable positional embeddings over $\boldsymbol{\delta}$, we view $[\widetilde{g}(x_i - x_j + \delta_x), \widetilde{g}(y_i - y_j + \delta_y)]$ as a continuous 2D index, and apply bilinear interpolation to continuously retrieve the positional embedding from $\mathbb{P} = \{\boldsymbol{p}_{[k,l]} | k, l = -\beta, \cdots, +\beta\}$, the dictionary of all positional embeddings at integer 2D indices (see Figure 3). Suppose the top-left corner nearest to the 2D coordinate has an integer index $[k, l]$, where $k = \lfloor \widetilde{g}(x_i - x_j + \delta_x) \rfloor$ and $l = \lfloor \widetilde{g}(y_i - y_j + \delta_y) \rfloor$ with the floor function $\lfloor \cdot \rfloor$. Then it yields the interpolated adversarial positional embedding below under the perturbation $\boldsymbol{\delta} = (\delta_x, \delta_y)$

$$\begin{aligned} \boldsymbol{p}_{[\widetilde{g}(x_i - x_j + \delta_x), \widetilde{g}(y_i - y_j + \delta_y)]} &= (1 - w_x)(1 - w_y)\boldsymbol{p}_{[k,l]} \\ &+ w_x(1 - w_y)\boldsymbol{p}_{[k+1,l]} + (1 - w_x)w_y\boldsymbol{p}_{[k,l+1]} \\ &+ w_x w_y \boldsymbol{p}_{[k+1,l+1]} \end{aligned}$$

with the interpolation weights

$$w_x = \widetilde{g}(x_i - x_j + \delta_x) - k, w_y = \widetilde{g}(y_i - y_j + \delta_y) - l.$$

The index function $\widetilde{g}$ and the coordinate perturbation (δ_x, δ_y) appear in the bilinear weights, where the back-propagated errors can go through $\widetilde{g}$ via these weights to update the perturbation.

Finally, the MIM objective with adversarial coordinates on relative positional embeddings becomes

$$\min_{\boldsymbol{\theta}, \boldsymbol{p}} \max_{\|\boldsymbol{\delta}\|_q \le \epsilon} \mathcal{L}(\{\boldsymbol{p}_{[\widetilde{g}(x_i - x_j + \delta_x), \widetilde{g}(y_i - y_j + \delta_y)]} | i \in \mathbb{M}\}; \boldsymbol{\theta})$$

with the constrained coordinate perturbation $\boldsymbol{\delta}$.

5 Experiments

In this section, we present the experiment results for the proposed AdPE method. We will show the results on Imagenet1K dataset with the pretrained model, and the transfer learning results on other datasets, as well as visualize the attention maps learned by the AdPE model.

Table 1: Comparison of our model with other methods on ViT-B and ViT-L. We evaluate them with the top-1 fine-tuning accuracy on ImageNet.

Method	Type	Extra Model	Epochs	ViT-B	ViT-L
supervised (He et al., 2022)	Supervised	-	300	82.3	82.6
MoCo-v3 (Chen et al., 2021)	Contrastive	momentum ViT	300	83.2	84.1
DINO (Caron et al., 2021)	Contrastive	momentum ViT	300	82.8	-
iBOT (Zhou et al., 2021)	Contrastive+MIM	momentum ViT	1600	84.0	84.8
BEiT (Bao et al., 2021)	MIM	DALLE+dVAE	800	83.2	85.2
data2vec (Baevski et al., 2022)	Contrastive	momentum ViT	800	84.2	86.2
CAE (Chen et al., 2022)	MIM	DALLE tokenizer	1600	83.9	86.3
SimMIM (Xie et al., 2022)	MIM	-	800	83.8	85.4
MaskFeat (Wei et al., 2022)	MIM	-	1600	84.0	85.7
MAE (He et al., 2022)	MIM	-	1600	83.6	85.9
MAE+ (ours)	MIM	-	1600	83.9	86.0
AdPE (ours)	MIM	-	1600	**84.4**	**86.3**

5.1 IMAGENET1K RESULTS

We adopt the MAE+ baseline presented in Section 3.2 as our baseline model for MIM pretraining. For the fair comparison, the same set of hyperparameters used in Appendix A are adopted. There are four combinations of design choices as discussed in Section 4 for the AdPE - two types of positional embeddings (absolute vs. relative), and two types of adversarial modes (embedding mode vs. coordinate mode). We follow the same evaluation protocol He et al. (2022) to report the results.

Table A.2 and Table A.3 in Appendix C shows the experiment results on Imagenet1K. The ViT-B with a 12-layer transformer encoder is pretrained over 400 epochs with an 8-layer transformer decoder. Then the pretrained backbone is fine-tuned end-to-end over 100 epochs with imagenet labels, and the top-1 accuracy is reported. For comparison, the MAE and MAE+ achieve 82.95% and 83.51% in top-1 accuracy, respectively.

From the ablation study in Table A.2, we can see that the AdPE with ℓ_∞-constraint has higher accuracy than that with ℓ_2-contraint for the same model type. Also, for the same type of positional embedding, the coordinate-mode adversaries perform better than the embedding-mode adversaries. We attribute this to the coordinate mode distorting image spatial structures in a more direct way with lower dimensionality (only two for x-and-y-axis) of adversaries than the embedding mode. This probably avoids the risk of over-distorting image structures arbitrarily with higher-dimensional adversaries in the embedding mode. Thus, a better balance is made to learn discriminative features from sufficiently adversarial rather than over-adversarial perturbations. The best accuracy for 400 epochs in Table A.2 is achieved by the coordinate-mode adversaries on RPE with $\epsilon = 20$ for the ℓ_∞-constraint, which we adopt in the following experiments.

In Table 1, we compare the AdPE with the other methods on Imagenet1K for pretraining ViT-B and ViT-L. The AdPE improves the accuracy of the MAE by 0.8% and 0.4% on ViT-B and ViT-L without using any external datasets or models. Its accuracy also is higher than that of the MAE+ by 0.4% and 0.3%.

We note that the AdPE is a pure MIM approach without contrastive pretraining or extra models like some other methods in Table 1. Particularly, contrastive pretraining often needs an additional momentum ViT branch, which makes it very slow and memory demanding in pretraining stage. For example, MoCo-v3 (Chen et al., 2021), a typical contrastive pretraining approach, used 128 V100 GPUs and took 10.24 GPU hours per epoch for pretraining, one order of magnitude slower than MAE and MAE+ that can be pretrained on merely eight V100 GPUs. Some approaches also resort to other models such as DALLE (Ramesh et al., 2021) as an extra tokenizer model. Although the AdPE is quite flexible to further improve these approaches by adding adversarial positional embeddings, it has already outperformed them.

Table 2: Transfer learning results on various downstream tasks.

Measurement	Epoch	ADE20K mIoU	COCO AP^{bbox}	 AP^{mask}
MoCo-v3 (Chen et al., 2021)	300	47.3	47.9	42.7
DINO (Caron et al., 2021)	400	47.2	-	-
BEiT (Bao et al., 2021)	800	47.1	49.8	44.4
CAE (Chen et al., 2022)	1600	50.2	50.0	44.0
iBOT (Zhou et al., 2021)	1600	50.0	51.2	44.2
SimMIM (Xie et al., 2022)	1600	50.0	49.1	43.8
MAE (He et al., 2022)	1600	48.1	50.3	44.9
AdPE (ours)	1600	**51.5**	**53.5**	**46.5**

5.2 Transfer Learning Results

We also conduct experiments on the transfer learning task to evaluate the generalization performance on ADE20K and COCO datasets. For a fair comparison, we still adopt the same protocol used in MAE to fine-tune the Mask R-CNN (He et al., 2017) end-to-end with the pretrained ViT-B backbone adapted for the FPN use on COCO (Lin et al., 2014), and report AP^{bbox} for object detection and AP^{mask} for instance segmentation. On ADE20K, we also follow the MAE by fine-tuning UpperNet (Xiao et al., 2018) for 100 epochs with a batch size of 16. The fine-tuning learning rate is set to 0.5 and $2e-4$ on COCO and ADE20K (Zhou et al., 2019), respectively.

The results in Table 2 show that across all tasks, our model performs the best among the compared methods, which significantly improves the SOTA approaches that even adopt extra datasets and/or models. This demonstrates its outstanding generalizability to other tasks.

5.3 Comparison with FGSM adversaries

We also extend the FGSM (Goodfellow et al., 2014) that applies instance-wise perturbations to image pixels. It follows the classic FGSM except the MAE reconstruction loss with the multi-crop tokenization in MAE+ are adopted to compute the adversarial perturbations. Table A.1 in Appendix B reports the fine-tuning results of the FGSM with various ϵ for ℓ_∞ constraint over 400 epochs of pretraining ViT-B. FGSM does not perform better than MAE+ (83.51%). This suggests that a straight extension of FGSM cannot improve the accuracy of a MIM-pretrained model in downstream tasks.

5.4 Visualization of Attention Maps

In Appendix D, we visualize the attention maps to compare MAE and AdPE. We show that AdPE does not focus its attention over local patches to infer missing ones. Instead, it is forced to explore non-local features in a larger spatial context. This verifies our assumption that the AdPE learns and integrates such high-level features from the global image context.

6 Conclusion

In this paper we present a new MAE+ baseline via multi-crop tokenization by extending the MAE-based masked image modeling. Upon the new baseline, we show that with Adversarial Positional Embeddings (AdPE), more discriminative features can be learned from the distorted image structures by preventing a pretrained vision transformer from simply using local correlations between patches to predict masked ones. This enables the transformer to learn high-level representations generalizable to downstream tasks. We impose coordinate-mode or embedding-mode adversaries on both absolute and relative positional embeddings, and the experiment results show that the AdPE with relative positional embeddings in the coordinate mode performs the best. We also show the AdPE has higher accuracy than the classic FGSM approach after fine-tuning the pretrained networks.

7 REPRODUCIBILITY

The source code of the proposed approach is available at `https://anonymous.4open.science/r/AdPE-ICLR`. Instructions to reproduce the reported results are included in the README.md file.

REFERENCES

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*, 2022.

Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9640–9649, 2021.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Qianjiang Hu, Xiao Wang, Wei Hu, and Guo-Jun Qi. Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1074–1083, 2021.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809, 2020.

Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. *arXiv preprint arXiv:2006.07589*, 2020.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. *Advances in Neural Information Processing Systems*, 32, 2019.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? *arXiv preprint arXiv:2106.11230*, 2021.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Xiao Wang, Yuhang Huang, Dan Zeng, and Guo-Jun Qi. Caco: Both positive and negative samples are directly learnable via cooperative-adversarial contrastive learning. *arXiv preprint arXiv:2203.14370*, 2022.

Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14668–14678, 2022.

Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10033–10041, 2021.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434, 2018.

Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9653–9663, 2022.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

A A NEW MIM BASELINE MAE+

In Section 3.2, we propose a new MIM baseline MAE+ by allowing multi-crop tokenization. Specifically, on Imagenet1K dataset, following random cropping and resizing , a $224 * 224$ input image is cropped into multiple $112 * 112$ smaller images.

The 75% of $7 * 7$ tokens resulting from each $112 * 112$ image with a grid of $16 * 16$ non-overlapping patches are randomly masked. Unlike the MAE, both masked and unmasked tokens will be fed through the MAE+ encoder.

For a fair comparison with MAE, the MAE+ adopts the same encoder-decoder architecture. Particularly, for the ViT-B backbone, the encoder of MAE+ is composed of 12 layers of transformers, and the decoder follows the same architecture in the MAE baseline with 8 transformer layers. The reconstruction loss over tokens is minimized to pretrain the network over various epochs. Then, the fine-tuning evaluation is made by re-training the whole network with Imagenet labels where a linear classification layer is added upon the average-pooled features from the pretrained ViT-B backbone. All experiments are run on a GPU server equipped with eight Nvidia V100 cards.

More specifically, in the pretraining stage, we follow the MAE baseline (He et al., 2022) without using color-jittering data augmentation, drop path or gradient clip. The xavier uniform is used to initialize all transformer blocks, and the same linear learning rate scaling rule is applied - the base learning rate of $1.5e-4$ is adapted with a batch size of 4096, yielding a learning rate of $lr = base_{lr} \times batchsize/256 = 2.4e-3$. AdamW optimizer is used with its momentum parameters of $\beta_1 = 0.9$ and $\beta_2 = 0.95$. The cosine decay is also adopted for scheduling learning rate with 40 warmup epochs and a weight decay of 0.05.

In the fine-tuning stage, we also use the common practice (He et al., 2022)(Bao et al., 2021) to supervise the end-to-end re-training of ViT-B. A learning rate of $4e-3$ is used with a batch size of 1024 and the cosine decay. AdamW is still used with the optimizer momentums set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$. A layer-wise learning rate decay of 0.75 is adopted. A total of 100 epochs is adopted for the fine-tuning with 5 warmup epochs. A drop path of 0.1 is aopted and RandAug (9, 0.5) is used for data augmentation along with mixup, cutmix and label smoothing.

B COMPARISON WITH FGSM ADVERSARIES

Table A.1: Top-1 accuracy of FGSM over 400 epochs of pretraining with ViT-B on Imagenet1k. The FGSM is imposed on the model pretrained with the MAE+ baseline.

	Cutoff $\epsilon = 0.001$		Cutoff $\epsilon = 0.01$		Cutoff $\epsilon = 0.1$	
Constraint	FT	Δ_{MAE+}	FT	Δ_{MAE+}	FT	Δ_{MAE+}
ℓ_∞	83.04	-0.47	83.09	-0.42	82.91	-0.60

The FGSM adversaries (Goodfellow et al., 2014) are one of the most classic instance-wise attacks on deep networks. It adds pixel-wise perturbations on the raw inputs that maximize the loss to make the worst-case attacks, and uses the resultant perturbed inputs to train the backbone network. The loss we adopt for MIM is the patch-wise reconstruction loss used in MAE (He et al., 2022). We then follow the same evaluation protocol to fine-tune the pretrained ViT-B backbone end-to-end with Imagenet1k labels. Table A.1 shows the results. We can see that the FGSM fails to improve the accuracy of the baseline MAE+ model. The results confirms the existing observation in literature (Tsipras et al., 2018) that the instance-wise perturbations cannot improve the standard accuracy for the downstream tasks.

Here we note that the goal of the AdPE is to improve the *standard* accuracy in downstream tasks since it is *not* an instance-wise perturbations sought by the FGSM imposed on raw images. Instead, the adversarial positional embeddings are a part of the pretrained transformer architecture. On the contrary, the FGSM-based adversaries are instance-wise attacks against the inputs, and they care more about the robust accuracy against various forms of adversarial perturbations. We would like to leave it to our future research to reveal if and how adversarial positional embeddings are related to the robustness against instance-wise attacks.

C Ablation Study of Various Design Choices

In this section, we report more experiment results. Table A.2 reports the top-1 accuracy over 400 epochs of pretraining ViT-B under different types of positional embeddings (PE), adversarial modes, constraints, and cutoff ϵ.

Table A.2: Top-1 fine-tuning (FT) accuracy over 400 epochs of pretraining ViT-B with different positional encodings (PE), adversarial modes (Mode), constraint types (Constraint), and the cutoff ϵ of constraint strength (Cutoff). It also gives the relative improvement over the baseline MAE model. Here, APE and RPE stand for absolute and relative positional embeddings, respectively.

PE	Mode	Constraint	Cutoff	FT	Δ_{MAE}	PE	Type	Constraint	Cutoff	ft	Δ_{MAE}
APE	Embed	ℓ_2	1	83.55	+0.60	RPE	Embed	ℓ_2	3	83.85	+0.90
APE	Embed	ℓ_2	3	83.60	+0.65	RPE	Embed	ℓ_2	5	83.85	+0.90
APE	Embed	ℓ_2	5	83.63	+0.68	RPE	Embed	ℓ_2	7	83.82	+0.87
APE	Embed	ℓ_∞	1	83.53	+0.58	RPE	Embed	ℓ_∞	10	83.92	+0.97
APE	Embed	ℓ_∞	3	**83.67**	+0.72	RPE	Embed	ℓ_∞	15	83.97	+1.02
APE	Embed	ℓ_∞	5	83.62	+0.67	RPE	Embed	ℓ_∞	20	**83.98**	+1.03
APE	Coord	ℓ_2	1	83.66	+0.71	RPE	Coord	ℓ_2	3	81.13	-1.82
APE	Coord	ℓ_2	3	83.54	+0.59	RPE	Coord	ℓ_2	5	81.18	-1.77
APE	Coord	ℓ_2	5	83.59	+0.64	RPE	Coord	ℓ_2	7	81.21	-1.74
APE	Coord	ℓ_∞	1	83.50	+0.55	RPE	Coord	ℓ_∞	10	83.73	+0.78
APE	Coord	ℓ_∞	3	**83.68**	+0.73	RPE	Coord	ℓ_∞	15	83.86	+0.91
APE	Coord	ℓ_∞	5	83.61	+0.66	RPE	Coord	ℓ_∞	20	**84.01**	+1.06

Table A.3 compares the AdPE with MAE+. It shows that the AdPE consistently improves over the MAE+ baseline – the coordinate-mode adversaries perform better than the embedding-mode adversaries, and the RPE outperforms the APE among the compared AdPE versions.

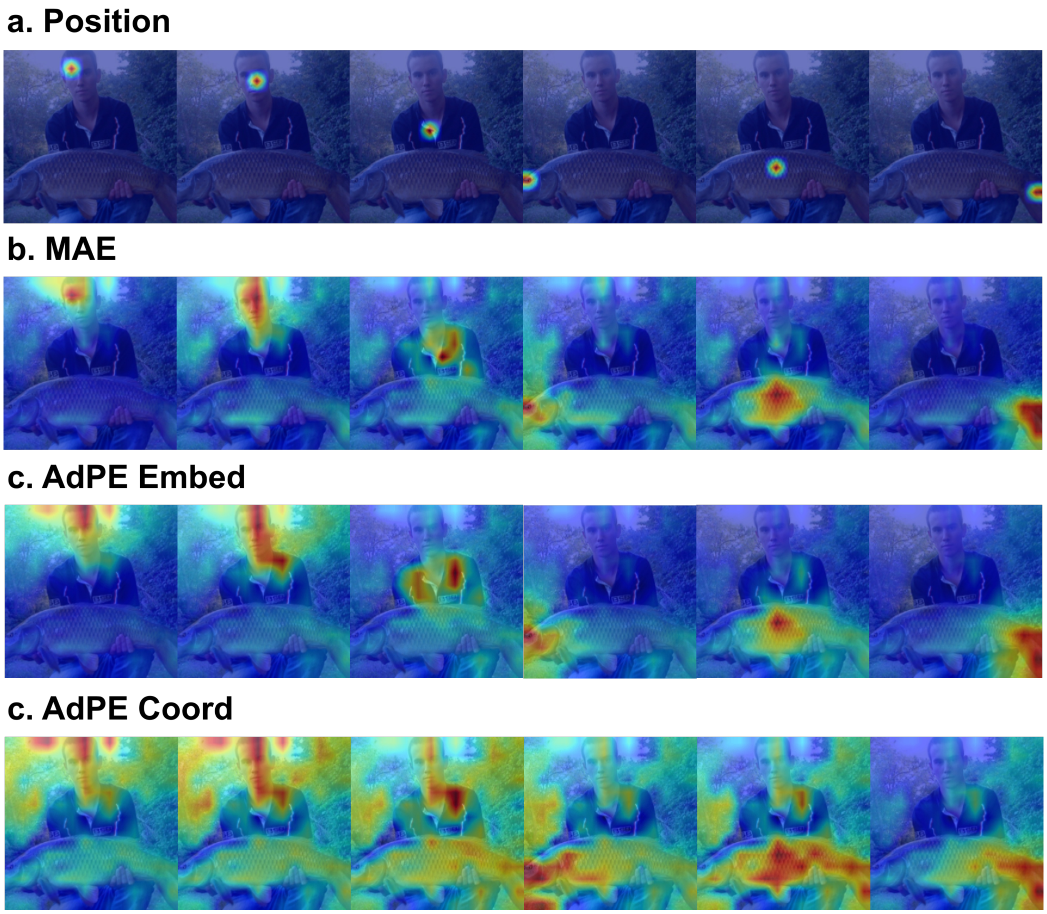
Table A.3: Comparison of the AdPE with MAE+. The results show that the AdPE makes consistent improvement (in Δ) over the MAE+ baseline.

			400ep		1600ep					400ep		1600ep	
Method	PE	Mode	FT	Δ	ft	Δ	Method	PE	Mode	FT	Δ	ft	Δ
MAE+	APE	-	83.51	-	83.94	-	MAE+	RPE	-	83.78	-	84.17	-
AdPE	APE	Embed	83.67	+0.16	84.08	+0.14	AdPE	RPE	Embed	83.98	+0.20	84.25	+0.08
AdPE	APE	Coord	83.68	+0.17	84.16	+0.22	AdPE	RPE	Coord	84.01	+0.23	84.36	+0.19

D Visualizing Attention Maps

In this section, we visualize attention maps when pretraining the AdPE. Figure A.1 visualizes attention maps of the first transformer encoder layer at different locations of an example image. It shows that compared to MAE, AdPE has a wider attention map covering large parts of the image. AdPE pretrained in coordinate-mode adversaries has largely distorted attention maps than that in embedding mode. It suggests that both AdPE models cannot simply use local correlations to infer missing patches. Instead, they are forced to explore those high-level features in a larger spatial context.

Figure A.2 visualizes the attention maps averaged over 1024 input images, which shows the same observation in the above.



a. Position

b. MAE

c. AdPE Embed

c. AdPE Coord

Figure A.1: The attention maps of the first transformer encoder layer. The MAE and AdPE are pretrained over 1600 epochs with the APE and $\epsilon = 3$ for the ℓ_∞ constraint.

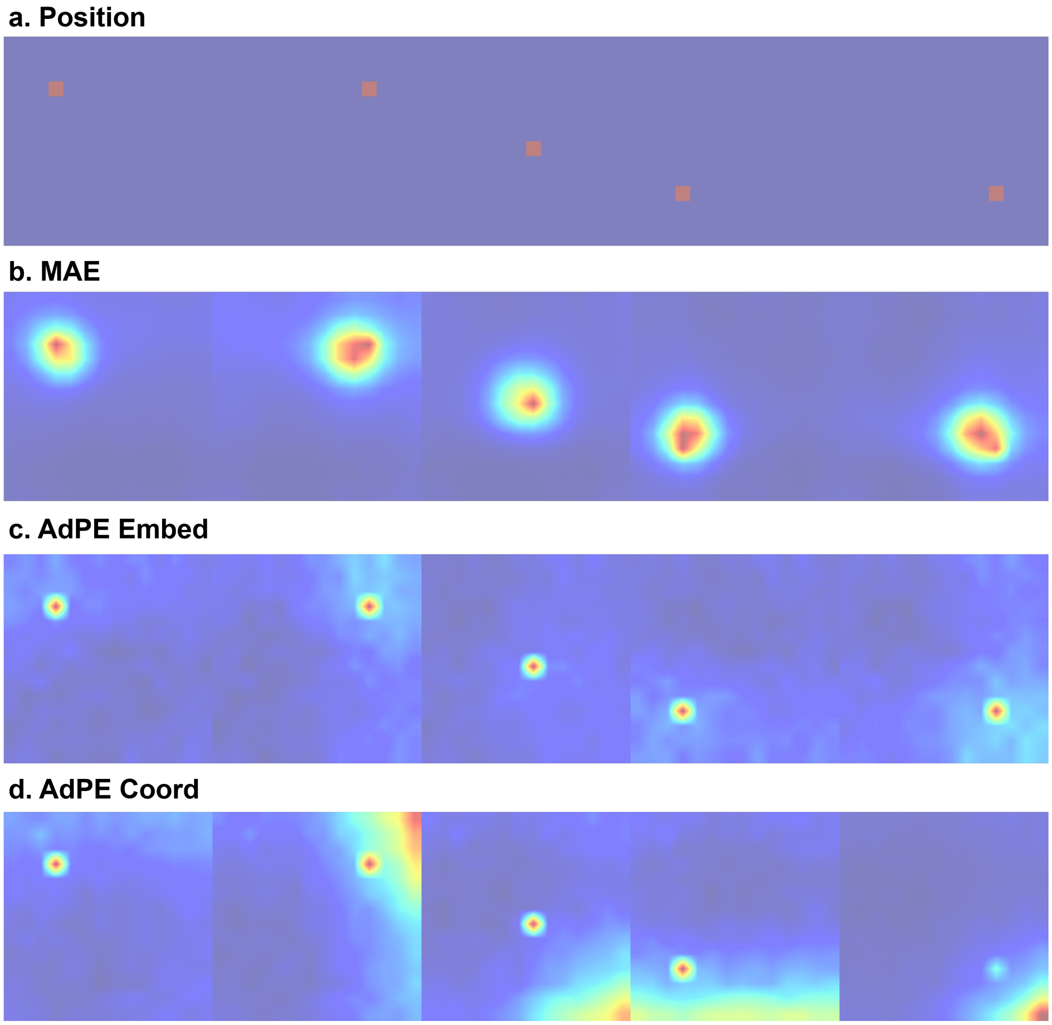


Figure A.2: The attention maps of the first transformer encoder layer averaged over 1024 input images. Again, the MAE and AdPE are pretrained over 1600 epochs with the APE and $\epsilon = 3$ for the ℓ_∞ constraint.