Scaling White-Box Transformers for Vision

Jinrui Yang[⋆1] **Xianhang Li**[⋆1] **Druv Pai**[2]
Yuyin Zhou[1] **Yi Ma**[2] **Yaodong Yu**[†2] **Cihang Xie**[†1]
[⋆]equal technique contribution, [†]equal advising
[1]UC Santa Cruz [2]UC Berkeley

Abstract

CRATE, a white-box transformer architecture designed to learn compressed and sparse representations, offers an intriguing alternative to standard vision transformers (ViTs) due to its inherent mathematical interpretability. Despite extensive investigations into the scaling behaviors of language and vision transformers, the scalability of CRATE remains an open question which this paper aims to address. Specifically, we propose CRATE-α, featuring strategic yet minimal modifications to the sparse coding block in the CRATE architecture design, and a light training recipe designed to improve the scalability of CRATE. Through extensive experiments, we demonstrate that CRATE-α can effectively scale with larger model sizes and datasets. For example, our CRATE-α-B substantially outperforms the prior best CRATE-B model accuracy on ImageNet classification by 3.7%, achieving an accuracy of 83.2%. Meanwhile, when scaling further, our CRATE-α-L obtains an ImageNet classification accuracy of 85.1%. More notably, these model performance improvements are achieved while preserving, and potentially even enhancing the interpretability of learned CRATE models, as we demonstrate through showing that the learned token representations of increasingly larger trained CRATE-α models yield increasingly higher-quality unsupervised object segmentation of images. The project page is https://rayjryang.github.io/CRATE-alpha/.

1 Introduction

Over the past several years, the Transformer architecture [42] has dominated deep representation learning for natural language processing (NLP), image processing, and visual computing [8, 2, 9, 5, 12]. However, the design of the Transformer architecture and its many variants remains largely empirical and lacks a rigorous mathematical interpretation. This has largely hindered the development of new Transformer variants with improved efficiency or interpretability. The recent white-box Transformer model CRATE [46] addresses this gap by deriving a simplified Transformer block via unrolled optimization on the so-called *sparse rate reduction* representation learning objective.

More specifically, layers of the white-box CRATE architecture are mathematically derived and fully explainable as unrolled gradient descent-like iterations for optimizing the sparse rate reduction. The self-attention blocks of CRATE explicitly conduct compression via denoising features against learned low-dimensional subspaces, and the MLP block is replaced by an incremental sparsification (via ISTA [1, 11]) of the features. As shown in previous work [47], besides mathematical interpretability, the learned CRATE models and features also have much better semantic interpretability than conventional transformers, i.e., visualizing features of an image naturally forms a zero-shot image segmentation of that image, even when the model is only trained on classification.

Scaling model size is widely regarded as a pathway to improved performance and emergent properties [44, 40, 41, 14]. Until now, the deployment of CRATE has been limited to relatively modest scales. The most extensive model described to date is the base model size encompasses 77.6M parameters

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-1167

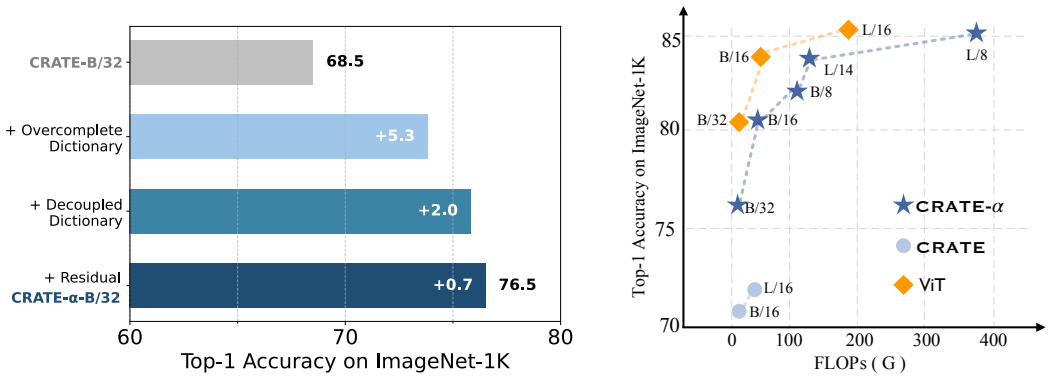

Figure 1: *(Left)* We demonstrate how modifications to the components enhance the performance of the CRATE model. The four models are trained using the same setup: first pre-trained on ImageNet-21K and then fine-tuned on ImageNet-1K. Details are provided in Section 3. *(Right)*. We compare the FLOPs and accuracy on ImageNet-1K of our methods with ViT [9] and CRATE [46]. The values of CRATE-α model correspond to those presented in Table 1. A more detailed comparison between CRATE-α and ViT is included in Appendix A.2.

(CRATE-Large) [46]. This contrasts sharply with standard Vision Transformers (ViTs [9]), which have been effectively scaled to a much larger model size, namely 22B parameters [5].

To this end, this paper provides the first exploration of training CRATE at different scales for vision, i.e., Tiny, Small, Base, Large, Huge. Detailed model specifications are given in Table 7 of Appendix A.1. To achieve effective scaling, we make two key changes. First, we identify the vanilla ISTA block within CRATE as a limiting factor that hinders further scaling. To overcome this, we significantly expand the channels, decouple the association matrix, and add a residual connection, resulting in a new model variant — CRATE-α. It is worth noting that this architecture change still preserves the mathematical interpretability of the model. Second, we propose an improved training recipe, inspired by previous work [38, 46, 39], for better coping the training with our new CRATE-α architecture.

We provide extensive experiments supporting the effective scaling of our CRATE-α models. For example, we scale the CRATE-α model from Base to Large size for supervised image classification on ImageNet-21K [6], achieving *85.1% top-1 accuracy on ImageNet-1K* at the Large model size. We further scale the model size from Large to Huge, utilizing vision-language pre-training with contrastive learning on DataComp1B [10], and achieve *a zero-shot top-1 accuracy of 72.3% on ImageNet-1K* at the Huge model size.[1] These results demonstrate the strong scalability of the CRATE-α model, shedding light on scaling up mathematically interpretable models for future work.

The main contributions of this paper are threefold:

1. We design three strategic yet minimal modifications for the CRATE model architecture to unleash its potential. In Figure 1, we reproduce the results of the CRATE model within our training setup, initially pre-training on ImageNet-21K classification and subsequently fine-tuning on ImageNet-1K classification. Compared to the vanilla CRATE model that achieves 68.5% top-1 classification accuracy on ImageNet-1K, our CRATE-α-B/32 model significantly improves the vanilla CRATE model by 8%, which clearly demonstrates the benefits of the three modifications to the existing CRATE model. Moreover, following the settings of the best CRATE model and changing the image patch size from 32 to 8, our CRATE-α-B model attains a top-1 accuracy of 83.2% on ImageNet-1K, exceeding the previous best CRATE model's score of 79.5% by a significant margin of 3.7%.
2. Through extensive experiments, we show that one can effectively scale CRATE-α via model size and data simultaneously. In contrast, when increasing the CRATE model from Base to Large model size, there is a marginal improvement on top-1 classification accuracy (+0.5%, from 70.8% to 71.3%) on ImageNet-1K, indicating diminished returns [46]. Furthermore, by scaling the training dataset, we achieved a substantial 1.9% improvement in top-1 classification accuracy on ImageNet-1K, increasing from 83.2% to 85.1% when going from CRATE-α Base to Large.
3. We further successfully scale CRATE-α model from Large to Huge by leveraging vision-language pre-training on DataComp1B. Compared to the Large model, the Huge model (CRATE-α-H) achieves a zero-shot top-1 classification accuracy of 72.3% on ImageNet-1K, marking a significant

[1]Model configurations are detailed in Table 7 (in Appendix A.1).

https://doi.org/10.52202/079017-1167

scaling gain of 2.5% over the Large model. These results indicate that the CRATE architecture has the potential to serve as an effective backbone for vision-language foundation models.

Related Work

White-box Transformers. [46, 45] argued that the quality of a learned representation can be assessed through a unified objective function called the *sparse rate reduction*. Based on this framework, [46, 45] developed a family of transformer-like deep network architectures, named CRATE, which are mathematically fully interpretable. CRATE models has been demonstrably effective on various tasks, including vision self-supervised learning and language modeling [26, 45]. Nevertheless, it remains unclear whether CRATE can scale as effectively as widely used black-box transformers. Previous work [46] suggests that scaling the vanilla CRATE model can be notably challenging.

Scaling ViT. ViT [9] represents the initial successful applications of Transformers to the image domain on a large scale. Many works [12, 31, 33, 32, 5, 37, 21, 22, 29, 18, 49] have deeply explored various ways of scaling ViTs in terms of model size and data size. From the perspective of self-supervision, MAE [12] provides a scalable approach to effectively training a ViT-Huge model using only ImageNet-1K. Following the idea of MAE, [31] further scales both model parameters to billions and data size to billions of images. Additionally, CLIP was the first to successfully scale ViT on a larger data scale (i.e., 400M) using natural language supervision. Based on CLIP, [32, 33] further scale the model size to 18 billion parameters, named EVA-CLIP-18B, achieving consistent performance improvements with the scaling of ViT model size. From the perspective of supervised learning, [49, 5] present a comprehensive analysis of the empirical scaling laws for vision transformers on image classification tasks, sharing some similar conclusions with [15]. [49] suggests that the performance-compute frontier for ViT models, given sufficient training data, tends to follow a saturating power law. More recently, [5] scales up ViT to 22 billion parameters. Scaling up different model architectures is non-trivial. [37, 21, 22] have made many efforts to effectively scale up different architectures. In this paper, due to the lack of study on the scalability of white-box models, we explore key architectural modifications to effectively scale up white-box transformers in the image domain.

2 Background and Preliminaries

In this section, we present the background on white-box transformers proposed in [46], including representation learning objectives, unrolled optimization, and model architecture. We first introduce the notation that will be used in the later presentation.

Notation. We use notation and problem setup following Yu et al. [46]. We use the matrix-valued random variable $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$ to represent the data, where each $\boldsymbol{x}_i \in \mathbb{R}^D$ is a "token", such that each data point is a realization of $\boldsymbol{X}$. For instance, $\boldsymbol{X}$ can represent a collection of image patches for an image, and $\boldsymbol{x}_i$ is the i-th image patch. We use $f \in \mathcal{F} \colon \mathbb{R}^{D \times N} \to \mathbb{R}^{d \times N}$ to denote the mapping induced by the transformer, and we let $\boldsymbol{Z} = f(\boldsymbol{X}) = [\boldsymbol{z}_1, \dots, \boldsymbol{z}_N] \in \mathbb{R}^{d \times N}$ denote the features for input data $\boldsymbol{X}$. Specifically, $\boldsymbol{z}_i \in \mathbb{R}^d$ denotes the feature of the i-th input token $\boldsymbol{x}_i$. The transformer f consists of multiple, say L, layers, and so can be written as $f = f^L \circ \cdots \circ f^1 \circ f^{\text{pre}}$, where $f^\ell \colon \mathbb{R}^{d \times N} \to \mathbb{R}^{d \times N}$ denotes the ℓ-th layer of the transformer, and the pre-processing layer is denoted by $f^{\text{pre}} = \mathbb{R}^{D \times N} \to \mathbb{R}^{d \times N}$. The input to the ℓ-th layer f^ℓ of the transformer is denoted by $\boldsymbol{Z}^\ell = \left[\boldsymbol{z}_1^\ell, \dots, \boldsymbol{z}_N^\ell\right] \in \mathbb{R}^{d \times N}$, so that $f^\ell \colon \boldsymbol{Z}^\ell \mapsto \boldsymbol{Z}^{\ell+1}$. In particular, $\boldsymbol{Z}^1 = f^{\text{pre}}(\boldsymbol{X}) \in \mathbb{R}^{d \times N}$ denotes the output of the pre-processing layer and the input to the first layer.

2.1 Sparse Rate Reduction

Following the framework proposed in [45], we posit that the goal of representation learning is to learn a feature mapping or *representation* $f \in \mathcal{F} \colon \mathbb{R}^{D \times N} \to \mathbb{R}^{d \times N}$ that transforms the input data $\boldsymbol{X}$ (which may have a nonlinear, multi-modal, and otherwise complicated distribution) into *structured and compact* features $\boldsymbol{Z}$, such that the token features lie on a union of low-dimensional subspaces, say with orthonormal bases $\boldsymbol{U}_{[K]} = (\boldsymbol{U}_k)_{k \in [K]} \in (\mathbb{R}^{d \times p})^K$. [46] proposes the ***S**parse **R**ate **R**eduction* (SRR) *objective* to measure the goodness of such a learned representation:

$$\max_{f \in \mathcal{F}} \mathbb{E}_{\boldsymbol{Z}=f(\boldsymbol{X})} \left[L_{\texttt{srr}}(\boldsymbol{Z}) \right] = \min_{f \in \mathcal{F}} \mathbb{E}_{\boldsymbol{Z}=f(\boldsymbol{X})} \left[R^c(\boldsymbol{Z} \,|\, \boldsymbol{U}_{[K]}) - R(\boldsymbol{Z} \,|\, \boldsymbol{U}_{[K]}) + \lambda \|\boldsymbol{Z}\|_1 \right], \quad (1)$$

https://doi.org/10.52202/079017-1167

where $\boldsymbol{Z} = f(\boldsymbol{X})$ denotes the token representation, $\|\boldsymbol{Z}\|_1$ denotes the ℓ^1 norm, and $R(\boldsymbol{Z})$, $R^c(\boldsymbol{Z} \mid \boldsymbol{U}_{[K]})$ are (estimates for) *rate distortions* [4, 7], defined as:

$$R(\boldsymbol{Z}) \doteq \frac{1}{2}\log\det\left(\boldsymbol{I} + \frac{d}{N\epsilon^2}\boldsymbol{Z}^\top\boldsymbol{Z}\right), \qquad R^c(\boldsymbol{Z} \mid \boldsymbol{U}_{[K]}) \doteq \sum_{k=1}^{K} R(\boldsymbol{U}_k^\top \boldsymbol{Z}). \tag{2}$$

In particular, $R^c(\boldsymbol{Z} \mid \boldsymbol{U}_{[K]})$ (resp. $R(\boldsymbol{Z})$) provide closed-form estimates for the number of bits required to encode the sample $\boldsymbol{Z}$ up to precision $\epsilon > 0$, conditioned (resp. unconditioned) on the samples being drawn from the subspaces with bases $\boldsymbol{U}_{[K]}$. Minimizing the term R^c improves the compression of the features $\boldsymbol{Z}$ against the posited model, and maximizing the term R promotes non-collapsed features. The remaining term $\lambda\|\boldsymbol{Z}\|_1$ promotes sparse features. Refer to [45] for more details about the desiderata and objective of representation learning via the rate reduction approach.

2.2 CRATE: Coding RATE Transformer

Unrolled optimization. To optimize the learning objective and learn compact and structured representation, one approach is unrolled optimization [11, 36]: each layer of the deep network implements an iteration of an optimization algorithm on the learning objective. For example, one can design the layer f^ℓ such that the forward pass is equivalent to a proximal gradient descent step for optimizing learning objective $L(\boldsymbol{Z})$, i.e., $\boldsymbol{Z}^{\ell+1} = f^\ell(\boldsymbol{Z}^\ell) = \texttt{Prox}[\boldsymbol{Z}^\ell - \eta \cdot \nabla_{\boldsymbol{Z}} L(\boldsymbol{Z}^\ell)]$. Here we use η to denote the step size and $\texttt{Prox}[\cdot]$ to denote the proximal operator [27].

One layer of the CRATE model. We now present the design of each layer of the white-box transformer architecture – Coding RATE Transformer (CRATE) – proposed in [46]. Each layer of CRATE contains two blocks: the compression block and the sparsification block. These correspond to a two-step alternating optimization procedure for optimizing the sparse rate reduction objective (1). Specifically, the ℓ-th layer of CRATE is defined as

$$\boldsymbol{Z}^{\ell+1} = f^\ell(\boldsymbol{Z}^\ell) = \texttt{ISTA}(\boldsymbol{Z}^{\ell+1/2} \mid \boldsymbol{D}^\ell), \quad \text{where} \quad \boldsymbol{Z}^{\ell+1/2} = \boldsymbol{Z}^\ell + \texttt{MSSA}(\boldsymbol{Z}^\ell). \tag{3}$$

Compression block (MSSA). The compression block in CRATE, called **M**ulti-head **S**ubspace **S**elf-**A**ttention block (MSSA), is derived for compressing the token set $\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N]$ by optimizing the compression term R^c (defined Eq. (1)), i.e.,

$$\boldsymbol{Z}^{\ell+1/2} = \boldsymbol{Z}^\ell + \texttt{MSSA}(\boldsymbol{Z}^\ell \mid \boldsymbol{U}_{[K]}^\ell) \approx \boldsymbol{Z}^\ell - \kappa\nabla_{\boldsymbol{Z}} R^c(\boldsymbol{Z}^\ell \mid \boldsymbol{U}_{[K]}^\ell), \tag{4}$$

where $\boldsymbol{U}_{[K]}^\ell$ denotes the (local) signal model at layer ℓ, and the MSSA operator is defined as

$$\texttt{MSSA}(\boldsymbol{Z} \mid \boldsymbol{U}_{[K]}) = \frac{\kappa p}{N\epsilon^2}[\boldsymbol{U}_1 \ \cdots \ \boldsymbol{U}_K] \begin{bmatrix} (\boldsymbol{U}_1^\top \boldsymbol{Z})\,\mathrm{softmax}((\boldsymbol{U}_1^\top \boldsymbol{Z})^\top (\boldsymbol{U}_1^\top \boldsymbol{Z})) \\ \vdots \\ (\boldsymbol{U}_K^\top \boldsymbol{Z})\,\mathrm{softmax}((\boldsymbol{U}_K^\top \boldsymbol{Z})^\top (\boldsymbol{U}_K^\top \boldsymbol{Z})) \end{bmatrix}. \tag{5}$$

Compared with the commonly used attention block in transformer [42], where the k-th attention head is defined as $(\boldsymbol{V}_k^\top \boldsymbol{Z})\,\mathrm{softmax}((\boldsymbol{Q}_k^\top \boldsymbol{Z})^\top (\boldsymbol{K}_k^\top \boldsymbol{Z}))$, MSSA uses only one matrix to obtain the query, key, and value matrices in the attention: that is, $\boldsymbol{U}_k = \boldsymbol{Q}_k = \boldsymbol{K}_k = \boldsymbol{V}_k$.

Sparse coding block (ISTA). The Iterative Shrinkage-Thresholding Algorithm (ISTA) block is designed to optimize the sparsity term and the global coding rate term, $\lambda\|\boldsymbol{Z}\|_0 - R(\boldsymbol{Z} \mid \boldsymbol{U}_{[K]})$ in (1). [46] shows that an optimization strategy for these terms posits a (complete) incoherent dictionary $\boldsymbol{D}^\ell \in \mathbb{R}^{d\times d}$ and takes a proximal gradient descent step towards solving the associated LASSO problem $\arg\min_{\boldsymbol{Z}\geq\boldsymbol{0}}[\frac{1}{2}\|\boldsymbol{Z}^{\ell+1/2} - \boldsymbol{D}^\ell\boldsymbol{Z}\|_2^2 + \lambda\|\boldsymbol{Z}\|_1]$, obtaining the iteration

$$\boldsymbol{Z}^{\ell+1} = \texttt{ISTA}(\boldsymbol{Z}^{\ell+1/2} \mid \boldsymbol{D}^\ell) = \mathrm{ReLU}(\boldsymbol{Z}^{\ell+1/2} + \eta\,(\boldsymbol{D}^\ell)^\top(\boldsymbol{Z}^{\ell+1/2} - \boldsymbol{D}^\ell\boldsymbol{Z}^{\ell+1/2}) - \eta\lambda\boldsymbol{1}). \tag{6}$$

In particular, the ISTA block sparsifies the intermediate iterates $\boldsymbol{Z}^{\ell+1/2}$ w.r.t. $\boldsymbol{D}^\ell$ to obtain $\boldsymbol{Z}^{\ell+1}$.

3 CRATE-α Model

In this section, we present the CRATE-α architecture, which is a variant of CRATE [46]. As shown in Fig. 1 (*Right*), there is a significant performance gap between the white-box transformer CRATE-B/16

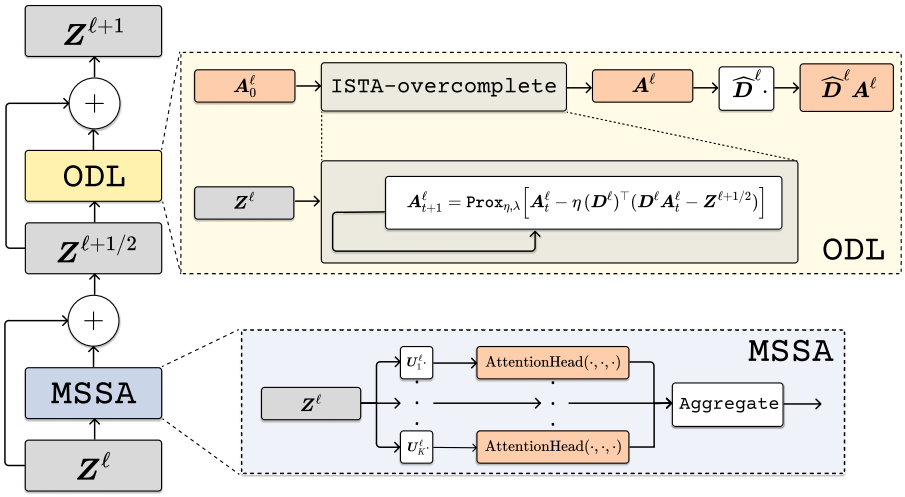

Figure 2: One layer of the CRATE-α model architecture. `MSSA` (**M**ulti-head **S**ubspace **S**elf-**A**ttention, defined in (5)) represents the compression block, and `ODL` (**O**vercomplete **D**ictionary **L**earning, defined in (12)) represents the sparse coding block. A more detailed illustration of the modifications is provided in Fig. 6 in the Appendix .

(70.8%) and the vision transformer ViT-B/16 (84.0%) [9]. One possible reason is that the `ISTA` block applies a complete dictionary $\boldsymbol{D} \in \mathbb{R}^{d\times d}$, which may limit its expressiveness. In contrast, the MLP block in the transformer[2] applies two linear transformations $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{d\times 4d}$, leading to the MLP block having 8 times more parameters than the `ISTA` block.

Since the ISTA block in CRATE applies a single incremental step to optimize the sparsity objective, applying an orthogonal dictionary can make it ineffective in sparsifying the token representations. Previous work [28] has theoretically demonstrated that overcomplete dictionary learning enjoys a favorable optimization landscape. In this work, we use an overcomplete dictionary in the sparse coding block to promote sparsity in the features. Specifically, instead of using a complete dictionary $\boldsymbol{D}^\ell \in \mathbb{R}^{d\times d}$, we use an overcomplete dictionary $\boldsymbol{D}^\ell \in \mathbb{R}^{d\times(Cd)}$, where $C > 1$ (a positive integer) is the overcompleteness parameter. Furthermore, we explore two additional modifications to the sparse coding block that lead to improved performance for CRATE. We now describe the three variants of the sparse coding block that we use in this paper.

Modification #1: Overparameterized sparse coding block. For the output of the ℓ-th CRATE attention block $\boldsymbol{Z}^{\ell+1/2}$, we propose to sparsify the token representations with respect to an overcomplete dictionary $\boldsymbol{D}^\ell \in \mathbb{R}^{d\times(Cd)}$ by optimizing the following LASSO problem,

$$\boldsymbol{A}^\ell \approx \arg\min_{\boldsymbol{A}\geq \boldsymbol{0}} \Big[\frac{1}{2}\|\boldsymbol{Z}^{\ell+1/2} - \boldsymbol{D}^\ell \boldsymbol{A}\|_2^2 + \lambda\|\boldsymbol{A}\|_1\Big]. \tag{7}$$

To approximately solve (7), we apply two steps of proximal gradient descent, i.e.,

$$\boldsymbol{A}_0^\ell = \boldsymbol{0}, \qquad \boldsymbol{A}_1^\ell = \texttt{Prox}_{\eta,\lambda}\big[\boldsymbol{A}_0^\ell; \boldsymbol{D}^\ell, \boldsymbol{Z}^{\ell+1/2}\big], \qquad \boldsymbol{A}_2^\ell = \texttt{Prox}_{\eta,\lambda}\big[\boldsymbol{A}_1^\ell; \boldsymbol{D}^\ell, \boldsymbol{Z}^{\ell+1/2}\big], \tag{8}$$

where `Prox` is the proximal operator of the above non-negative LASSO problem (7) and defined as

$$\texttt{Prox}_{\eta,\lambda}[\boldsymbol{A}; \boldsymbol{D}, \boldsymbol{Z}] = \mathrm{ReLU}(\boldsymbol{A} - \eta\boldsymbol{D}^\top(\boldsymbol{D}\boldsymbol{A} - \boldsymbol{Z}) - \eta\lambda\boldsymbol{1}). \tag{9}$$

The output of the sparse coding block is defined as

$$\boldsymbol{Z}^{\ell+1} = \boldsymbol{D}^\ell \boldsymbol{A}^\ell, \quad \text{where} \quad \boldsymbol{A}^\ell = \boldsymbol{A}_2^\ell \doteq \texttt{ISTA-OC}(\boldsymbol{Z}^{\ell+1/2} \mid \boldsymbol{D}^\ell). \tag{10}$$

Namely, $\boldsymbol{A}^\ell$ is a sparse representation of $\boldsymbol{Z}^{\ell+1/2}$ with respect to the overcomplete dictionary $\boldsymbol{D}^\ell$. The original CRATE `ISTA` tries to learn a complete dictionary $\boldsymbol{D} \in \mathbb{R}^{d\times d}$ to transform and sparsify the features $\boldsymbol{Z}$. By leveraging more atoms than the ambient dimension, the overcomplete dictionary $\boldsymbol{D} \in \mathbb{R}^{d\times(Cd)}$ can provide a redundant yet expressive codebook to identify the salient sparse

[2]The MLP block is defined as $\boldsymbol{Z}^{\ell+1} = \boldsymbol{Z}^\ell + \boldsymbol{W}_2\sigma(\boldsymbol{W}_1^\top \boldsymbol{Z}^{\ell+1/2})$, where σ is the nonlinear activation function and $\boldsymbol{Z}^{\ell+1/2}$ denotes the output of the attention block.

https://doi.org/10.52202/079017-1167

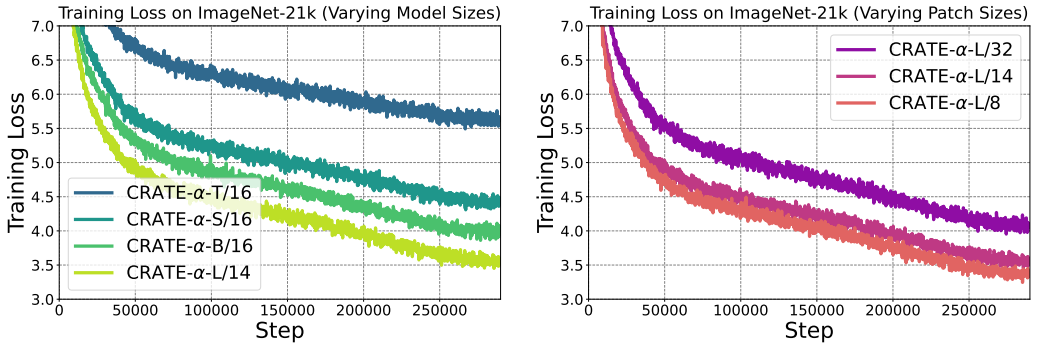

Figure 3: Training loss curves of CRATE-α on ImageNet-21K. (*Left*) Comparing training loss curves across CRATE-α with different model sizes. (*Right*) Comparing training loss curves across CRATE-α-Large with different patch sizes.

structures underlying $\boldsymbol{Z}$. As shown in Fig. 1, the overcomplete dictionary design leads to 5.3% improvement compared to the vanilla CRATE model.

Modification #2: Decoupled dictionary. We propose to apply a decoupled dictionary $\widehat{\boldsymbol{D}}^\ell$ in the last step (defined in (10) of the sparse coding block, $\boldsymbol{Z}^{\ell+1} = \widehat{\boldsymbol{D}}^\ell \boldsymbol{A}^\ell$, where $\widehat{\boldsymbol{D}}^\ell \in \mathbb{R}^{d\times(Cd)}$ is a different dictionary compared to $\boldsymbol{D}^\ell$. By introducing the decoupled dictionary, we further improve the model performance by 2.0%, as shown in Fig. 1. We denote this mapping from $\boldsymbol{Z}^{\ell+1/2}$ to $\boldsymbol{Z}^{\ell+1}$ as the **O**vercomplete **D**ictionary **L**earning block (ODL), defined as follows:

$$\texttt{ODL}(\boldsymbol{Z}^{\ell+1/2} \mid \boldsymbol{D}^\ell, \widehat{\boldsymbol{D}}^\ell) \doteq \widehat{\boldsymbol{D}}^\ell \cdot \texttt{ISTA-OC}(\boldsymbol{Z}^{\ell+1/2} \mid \boldsymbol{D}^\ell) = \widehat{\boldsymbol{D}}^\ell \boldsymbol{A}^\ell. \tag{11}$$

Modification #3: Residual connection. Based on the previous two modifications, we further add a residual connection, obtaining the following modified sparse coding block:

$$\boldsymbol{Z}^{\ell+1} = \boldsymbol{Z}^{\ell+1/2} + \texttt{ODL}(\boldsymbol{Z}^{\ell+1/2} \mid \boldsymbol{D}^\ell, \widehat{\boldsymbol{D}}^\ell). \tag{12}$$

An intuitive interpretation of this modified sparse coding block is as follows: instead of directly sparsifying the feature representations $\boldsymbol{Z}$, we first identify the potential sparse patterns present in $\boldsymbol{Z}$ by encoding it over a learned dictionary. Subsequently, we incrementally refine $\boldsymbol{Z}$ by exploiting the sparse codes obtained from the previous encoding step. From Fig. 1, we find that the residual connection leads to a 0.7% improvement.

To summarize, to effectively scale white-box transformers, we implement three modifications to the vanilla white-box CRATE model proposed in [46]. Specifically, in our CRATE-α model, we introduce a decoupling mechanism, quadruple the dimension of the dictionary ($4\times$), and incorporate a residual connection in the sparse coding block.

4 Experiments

Overall. The experimental section consists of three parts: (1) **Scaling study:** We thoroughly investigate the scaling behaviors of CRATE-α from Base to Large size and ultimately to Huge size. (2) **Downstream applications:** To further verify the broader benefits of scaling the CRATE-α model, we conduct additional experiments on real-world downstream tasks and present preliminary exploration results of CRATE-α on language tasks. (3) **Interpretability:** In addition to scalability, we study the interpretability of CRATE-α across different model sizes.

4.1 Dataset and Evaluation

Scaling Study. For the transition from Base to Large size, we pre-train our model on ImageNet-21K and fine-tune it on ImageNet-1K via supervised learning. When scaling from Large to Huge, we utilize the DataComp1B [10] dataset within a vision-language pre-training paradigm, allowing us to study the effects of scaling the model to a massive size. For evaluation, we evaluate the zero-shot accuracy of these models on ImageNet-1K.

Table 1: Top-1 accuracy of CRATE-α on ImageNet-1K with different model scales when pre-trained on ImageNet-21K and then fine-tuned on ImageNet-1K. For comparison, we also list the results from the paper [46] which demonstrate the diminished return from CRATE base to large, trained only on ImageNet-1K. "IN-21K" refers to ImageNet-21K. (‡Results from [46].)

Models (Base)	ImageNet-1K(%)	Models (Large)	ImageNet-1K(%)
CRATE-B/16 w/o IN-21K	70.8‡	CRATE-L/16 w/o IN-21K	71.3‡
CRATE-α-B/32	76.5	CRATE-α-L/32	80.2
CRATE-α-B/16	81.2	CRATE-α-L/14	83.9
CRATE-α-B/8	83.2	CRATE-α-L/8	85.1

Downstream Applications. We include additional experimental results on four downstream datasets (CIFAR-10/100, Oxford Flowers, and Oxford-IIT Pets). We also examine the dense prediction capability of CRATE-α by training it on segmentation tasks using the ADE20K dataset [51]. For language tasks, we conduct new experiments with CRATE-α using autoregressive training on OpenWebText, following the setup in nanoGPT [16].

Interpretability. Following the evaluation setup of CRATE as outlined in [46], we apply MaskCut [43] to validate and evaluate the rich semantic information captured by our model in a zero-shot setting, including both qualitative and quantitative results.

4.2 Training and Fine-tuning Procedures

Scaling Study. (1) Base to Large size: We initially pre-train the CRATE-α model on ImageNet-21K and subsequently fine-tune it on ImageNet-1K. During the pre-training phase, we set the learning rate to 8×10^{-4}, weight decay to 0.1, and batch size to 4096. We apply data augmentation techniques such as Inception crop [35] resized to 224 and random horizontal flipping. In the fine-tuning phase, we adjust the base learning rate to 1.6×10^{-4}, maintain weight decay at 0.1, and batch size at 4096. We apply label smoothing with a smoothing parameter of 0.1 and apply data augmentation methods including Inception crop, random horizontal flipping, and random augmentation with two transformations (magnitude of 9). For evaluation, we resize the smaller side of an image to 256 while maintaining the original aspect ratio and then crop the central portion to 224×224. In both the pre-training and fine-tuning phases, we use the AdamW optimizer [24] and incorporate a warm-up strategy, characterized by a linear increase over 10 epochs. Both the pre-training and fine-tuning are conducted for a total of 91 epochs, utilizing a cosine decay schedule.

(2) Large to Huge size: In the pre-training stage, we utilize an image size of 84×84, and the maximum token length is 32, with a total of 2.56 billion training samples. During the fine-tuning stage, we increase the image size to 224×224 while maintaining the maximum token length at 32, with a 512 million training samples. Here, the key distinction between the pre-training stage and the fine-tuning stage is the image size. A smaller image size results in a faster training speed. In the configurations of CRATE-α-CLIPA-B, CRATE-α-CLIPA-L, and CRATE-α-CLIPA-H, we use the CRATE-α model as the vision encoder, and utilize the same pre-trained huge transformer model from CLIPA [18] as the text encoder. For both the pre-training and fine-tuning stages, we freeze the text encoder and only train the vision encoder, i.e., the CRATE-α model. As we will show in the later results, this setup effectively demonstrates the scaling behaviors of CRATE-α models in the image domain. Detailed hyperparameter settings can be found in Appendix A.

Downstream Applications. On four downstream datasets, we follow the training setup from [46]. For the segmentation task, we compare the performance of CRATE and CRATE-α on the ADE20K dataset, mainly following the setup of [30] with minor modifications. Our batch size is set to 128, and the total number of training steps is 5000. For the language task, we conduct experiments with CRATE-α using autoregressive training on OpenWebText, following the setup in [16]. We compare CRATE-α models (57M and 120M) with CRATE and GPT-2, using results from CRATE reported in [45].

https://doi.org/10.52202/079017-1167

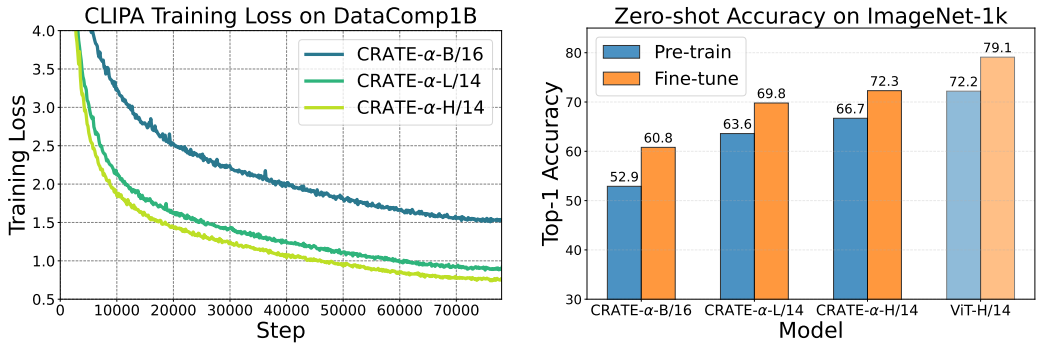

Figure 4: (*Left*) Comparing training loss curves of CRATE-α-CLIPA with different model sizes on DataComp1B. (*Right*) Comparing zero-shot accuracy of CRATE-α-B/L/H models and ViT-H on ImageNet-1K.

4.3 Results and Analysis

Scaling the CRATE-α Model from Base to Large. As shown in Table 1, we compare CRATE-α-B and CRATE-α-L at patch sizes 32, 16, and 8. Firstly, we find our proposed CRATE-α-L consistently achieves significant improvements across all patch sizes. Secondly, compared with the results of the vanilla CRATE (the first row of Table 1), increasing from CRATE-B to CRATE-L results in only a 0.5% improvement on ImageNet-1K. This indicates a case of diminishing returns. These findings compellingly highlight that the scalability of CRATE-α models significantly outperforms that of the vanilla CRATE. Meanwhile, the training loss in the pre-training stage is presented in Fig. 3; as the model capacity increases, the trend of the training loss improves predictably. This phenomenon is also described in [9].

Scaling the CRATE-α Model from Large to Huge. From the results shown in Fig. 4, we find that: (1) CRATE-α-CLIPA-L/14 significantly outperforms CRATE-α-CLIPA-B/16 by 11.3% and 9.0% in terms of ImageNet-1K zero-shot accuracy during the pre-training and fine-tuning stages, respectively. The substantial benefit suggests that the quality of learned representation may be constrained by the model size. Therefore, increasing the model size effectively leverages larger amounts of data. (2) When continuing to scale up model size, we also observe that CRATE-α-CLIP-H/14 continues to benefit from larger training datasets, outperforming CRATE-α-CLIP-L/14 by 3.1% and 2.5% in terms of ImageNet-1K zero-shot accuracy during the pre-training and fine-tuning stages, respectively. This demonstrates the strong scalability of the CRATE-α model. To explore the performance ceiling, we train a standard ViT-CLIPA-H/14 from scratch and observe improved performance.

Downstream Applications. On four downstream datasets, as shown in Table 2, we find that CRATE-α consistently outperforms CRATE, with both models pre-trained on IN21K, while CRATE-α demonstrates improved performance as model size increases. For the segmentation task, results in Table 3 show that CRATE-α consistently outperforms CRATE across all key metrics, with both models pre-trained on IN21K. These findings indicate significant performance gains in vision tasks beyond classification. For the language task, Table 4 shows that CRATE-α significantly improves over CRATE in language modeling. Due to limited time and resource constraints, we completed 80% of the total iterations for CRATE-α-small and 55% for CRATE-α-base, compared to the 600K total iterations used for CRATE. Nevertheless, CRATE-α still demonstrated notable improvements.

Interpretability. As shown in Fig. 5, we provide the segmentation visualization on COCO val2017 [20] for CRATE-α, CRATE, and ViT, respectively. We find that our model preserves and even improves the (semantic) interpretability advantages of CRATE. Moreover, we summarize quantitative evaluation results on COCO val2017 in Table 6. Interestingly, when scaling up model size for CRATE-α, the Large model improves over the Base model in terms of object detection and segmentation.

4.4 Compute-efficient Scaling Strategy

We further explore methods to scale models efficiently in terms of computation. Table 1 demonstrates that the CRATE-α model scales effectively from the Base model to its larger variants. However, the pre-training computation for the top-performing model, CRATE-α-L/8, is resource-intensive on

Table 2: The performance comparison between CRATE and CRATE-α across various datasets.

Dataset	CRATE-B/32	CRATE-α-B/32	CRATE-α-L/32	CRATE-α-B/16	CRATE-α-L/14
CIFAR-10	97.22	98.17	98.68	98.67	99.10
CIFAR-100	85.27	89.40	91.16	90.58	92.57
Oxford Flowers-102	93.90	97.77	99.01	99.27	99.56
Oxford-IIIT-Pets	80.38	88.19	90.46	92.70	93.98

Table 3: Performance comparison of CRATE models with different configurations.

Model	Scope	mIoU	mAcc	aAcc
CRATE-α-B/32	global	35.35	45.28	77.63
CRATE-B/32	global	30.28	39.29	75.21

Table 4: The comparison between CRATE and CRATE-α on the NLP task using the OpenWebText dataset.

	GPT-2-base	CRATE-base	CRATE-α-small	CRATE-α-base
Model size	124M	60M	57M	120M
Cross-entropy validation loss	2.85	3.37	3.28	3.14

Table 5: Compute-efficient scaling strategy. To reduce the compute requirements of the pre-training stage, we use a model with a larger patch size. This results in a shorter token length for the same input size. The second and fourth columns indicate the compute requirements for the pre-training and fine-tuning stages, respectively, measured in TPU v3 core-hours. Details are provided in Section 4.4.

Pre-train	Core-hours	Fine-tune	Core-hours	Total core-hours	IN-1K(%)
CRATE-α-L/32	2,652	CRATE-α-L/14	872	3,524	83.7
		CRATE-α-L/8	3,486	6,138	84.2
CRATE-α-L/14	8,947	CRATE-α-L/14	872	9,819	83.9
CRATE-α-L/8	35,511	CRATE-α-L/8	3,486	38,997	85.1

ImageNet-21K. Inspired by CLIPA [18], we aim to reduce computational demands by using reduced image token sequence lengths, while maintaining the same training setup during the fine-tuning stage. The results are summarized in Table 5.

Results and analysis. (1) When fine-tuning with CRATE-α-L/14 and using CRATE-α-L/32 for pre-training on ImageNet-21K, this approach consumes about 35% of the TPU v3 core-hours required by CRATE-α-L/14, yet achieves a promising 83.7% top-1 accuracy on ImageNet-1K, comparable to the 83.9% achieved by CRATE-α-L/14; (2) When fine-tuning with CRATE-α-L/8 and using CRATE-α-L/32 for pre-training, this approach consumes just 15% of the training time required by CRATE-α-L/8, yet it still achieves a promising 84.2% top-1 accuracy on ImageNet-1K, compared to 85.1% when using the CRATE-α-L/8 model in the pre-training stage; (3) While the total computational cost of CRATE-α-L/32 + CRATE-α-L/8 is less than that of CRATE-α-L/14 + CRATE-α-L/14, the performance of the former is slightly better. In summary, we find that this strategy offers a valuable reference for efficiently scaling CRATE-α models in the future.

5 Discussion

Limitations. Although we have used some existing compute-efficient training methods (e.g., CLIPA [18]) and have initiated an exploration into compute-efficient scaling strategies for white-box transformers in Section 4.4, this work still requires a relatively large amount of computational resources, which may not be easily accessible to many researchers.

Societal impact. A possible broader implication of this research is the energy consumption needed to conduct the experiments in our scaling study. However, there is growing interest in developing white-box transformers for better interpretability and transparency across a wide range of tasks and domains, including image segmentation [46], self-supervised masked autoencoders [26], and integrated sensing and communications [50], etc. Moreover, our results on the scalability of white-

https://doi.org/10.52202/079017-1167

Figure 5: **Visualization of segmentation on COCO val2017 [20] with MaskCut [43].** (*Top row*) Supervised CRATE-α effectively identifies the main objects in the image. Compared with CRATE (*Middle row*), CRATE-α achieves better segmentation performance in terms of boundary. (*Bottom row*) Supervised ViT fails to identify the main objects in most images. We mark failed image with □.

Table 6: **Object detection and fine-grained segmentation via MaskCut on COCO val2017 [20].** We evaluate models of various scales and assess their average precision using COCO's official evaluation metric. Compared with existing models such as CRATE and ViT, CRATE-α model achieves a notable performance gain. In addition, when scaling CRATE-α from base to large, it also exhibits the benefit of scalability.

		Detection			Segmentation		
Model	Train	AP_{50} ↑	AP_{75} ↑	AP ↑	AP_{50} ↑	AP_{75} ↑	AP ↑
CRATE-B/8 [47]	Supervised	2.9	1.0	1.3	2.2	0.7	1.0
ViT-B/8 [47]	Supervised	0.8	0.2	0.4	0.7	0.5	0.4
CRATE-α-B/8	Supervised	3.5	1.1	1.5	2.2	1.0	1.1
CRATE-α-L/8	Supervised	4.0	1.7	2.0	2.7	1.1	1.4

box transformers could also shed light on scaling up a broader class of white-box deep neural networks, such as white-box ISTA networks and their variants [11, 34, 3, 48, 17], designed via unrolled optimization. In summary, we believe that our findings and insights could be helpful for developing white-box transformers for a wide range of applications and tasks, benefiting a broad audience interested in building more interpretable and performant deep learning models and further amortizing the pre-training compute costs.

6 Conclusion

This paper provides the first exploration of training white-box transformer CRATE at scale for vision tasks. We introduce both principled architectural changes and improved training recipes to unleash the potential scalability of the CRATE type architectures. With these modifications, we successfully scale up the CRATE-α model along both the dimensions of model size and data size, while preserving, in most cases even improving, the semantic interpretability of the learned white-box transformer models. We believe this work provides valuable insights into scaling up mathematically interpretable deep neural networks, not limited to transformer-like architectures.

Acknowledgement

This work is supported by a gift from Open Philanthropy, TPU Research Cloud (TRC) program, and Google Cloud Research Credits program.

References

[1] Thomas Blumensath and Mike E Davies. Iterative thresholding for sparse approximations. *Journal of Fourier analysis and Applications*, 14:629–654, 2008.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ista and its practical weights and thresholds. *Advances in Neural Information Processing Systems*, 31, 2018.

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

[5] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.

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

[7] Harm Derksen, Yi Ma, Wei Hong, and John Wright. Segmentation of multivariate mixed data via lossy coding and compression. In *Visual Communications and Image Processing 2007*, volume 6508, pages 170–181. SPIE, 2007.

[8] 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.

[9] 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.

[10] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2024.

[11] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pages 399–406, 2010.

[12] 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*, pages 16000–16009, 2022.

[13] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip. July 2021.

[14] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[16] Andrej Karpathy. nanogpt, 2022. URL https://github.com/karpathy/nanoGPT. GitHub repository.

 https://doi.org/10.52202/079017-1167

[17] Mingyang Li, Pengyuan Zhai, Shengbang Tong, Xingjian Gao, Shao-Lun Huang, Zhihui Zhu, Chong You, Yi Ma, et al. Revisiting sparse convolutional model for visual recognition. *Advances in Neural Information Processing Systems*, 35:10492–10504, 2022.

[18] Xianhang Li, Zeyu Wang, and Cihang Xie. An inverse scaling law for clip training. *Advances in Neural Information Processing Systems*, 36, 2024.

[19] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *CVPR*, 2023.

[20] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[21] 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*, pages 10012–10022, 2021.

[22] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[23] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018.

[26] Druv Pai, Ziyang Wu, Sam Buchanan, Tianzhe Chu, Yaodong Yu, and Yi Ma. Masked completion via structured diffusion with white-box transformers. In *Conference on Parsimony and Learning (Recent Spotlight Track)*, 2023.

[27] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014.

[28] Qing Qu, Yuexiang Zhai, Xiao Li, Yuqian Zhang, and Zhihui Zhu. Geometric analysis of nonconvex optimization landscapes for overcomplete learning. In *International Conference on Learning Representations*, 2019.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[30] Sucheng Ren, Fangyun Wei, Zheng Zhang, and Han Hu. Tinymim: An empirical study of distilling mim pre-trained models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3687–3697, 2023.

[31] Mannat Singh, Quentin Duval, Kalyan Vasudev Alwala, Haoqi Fan, Vaibhav Aggarwal, Aaron Adcock, Armand Joulin, Piotr Dollár, Christoph Feichtenhofer, Ross Girshick, et al. The effectiveness of mae pre-pretraining for billion-scale pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5484–5494, 2023.

[32] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

[33] Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. Eva-clip-18b: Scaling clip to 18 billion parameters. *arXiv preprint arXiv:2402.04252*, 2024.

[34] Xiaoxia Sun, Nasser M Nasrabadi, and Trac D Tran. Supervised deep sparse coding networks. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 346–350. IEEE, 2018.

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[36] Bahareh Tolooshams and Demba Ba. Stable and interpretable unrolled dictionary learning. *arXiv preprint arXiv:2106.00058*, 2021.

[37] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34: 24261–24272, 2021.

[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. PMLR, 2021.

[39] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *European conference on computer vision*, pages 516–533. Springer, 2022.

[40] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[41] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[42] 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.

[43] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3124–3134, 2023.

[44] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

[45] Yaodong Yu, Sam Buchanan, Druv Pai, Tianzhe Chu, Ziyang Wu, Shengbang Tong, Hao Bai, Yuexiang Zhai, Benjamin D Haeffele, and Yi Ma. White-box transformers via sparse rate reduction: Compression is all there is? *arXiv preprint arXiv:2311.13110*, 2023.

[46] Yaodong Yu, Sam Buchanan, Druv Pai, Tianzhe Chu, Ziyang Wu, Shengbang Tong, Benjamin Haeffele, and Yi Ma. White-box transformers via sparse rate reduction. *Advances in Neural Information Processing Systems*, 36, 2023.

[47] Yaodong Yu, Tianzhe Chu, Shengbang Tong, Ziyang Wu, Druv Pai, Sam Buchanan, and Yi Ma. Emergence of segmentation with minimalistic white-box transformers. In *Conference on Parsimony and Learning*, pages 72–93. PMLR, 2024.

[48] John Zarka, Louis Thiry, Tomás Angles, and Stéphane Mallat. Deep network classification by scattering and homotopy dictionary learning. *arXiv preprint arXiv:1910.03561*, 2019.

[49] 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.

 https://doi.org/10.52202/079017-1167

[50] Bowen Zhang and Geoffrey Ye Li. White-box 3d-omp-transformer for isac. *arXiv preprint arXiv:2407.02251*, 2024.

[51] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

Appendix

A Additional Experiments and Details

A.1 Model configuration.

We provide details about CRATE-α model configurations in Table 7.

Table 7: Model configurations for different sizes of CRATE-α, parameter counts, and comparisons to CRATE models. L is depth, d is the hidden size, and K is the number of heads.

Model Size	L	d	K	CRATE-α # Params	CRATE # Params
Tiny	12	192	3	4.8M	1.7M
Small	12	576	12	41.0M	13.1M
Base	12	768	12	72.3M	22.8M
Large	24	1024	16	253.8M	77.6M
Huge	32	1280	16	526.8M	159.8M

Table 8: The comparison between CRATE-α and ViT. FLOPs and throughput are calculated based on an input size of 224x224 on an NVIDIA RTX A6000 graphics card.

Model	FLOPs (G)	#Params (M)	Throughput	Model	FLOPs (G)	#Params (M)	Throughput
CRATE-α-B/32	6.4	74.0	499	ViT-B/32	4.4	88.2	706
CRATE-α-B/16	25.8	72.3	233	ViT-B/16	17.6	86.5	375
CRATE-α-L/32	22.8	256.0	215	ViT-L/32	15.4	306.5	329
CRATE-α-L/14	119.7	253.7	56	ViT-L/14	81.1	304.1	85

A.2 Comparison of model structure with ViT.

We also compare CRATE-α to ViT in terms of computational costs, number of parameters, and inference speed. These comparisons are summarized in Table 8, where CRATE-α matches ViT's efficiency while achieving similar accuracy. With the same number of layers and embedding dimensions, CRATE-α has fewer parameters than ViT, and its FLOPs/Throughput is slightly higher.

To more accurately compare CRATE-α and ViT with larger model sizes, we conduct experiments on CRATE-α-L/16 with an image resolution of 336, nearly matching the setup of ViT-L/16. Both models use a similar amount of FLOPs: 210G for CRATE-α-L/16 compared to 191G for ViT-L/16. The throughput, or images processed per second, is also comparable at 35.53 for our model versus 35.56 for ViT-L/16. The accuracy of CRATE-α-L/16 reach 84.6%, closely approaching ViT's 85.2% under similar conditions. Meanwhile, combining the trend from Figure 1 (right) in the main paper, this narrowing performance gap from Base to Large model size suggests that CRATE-α can nearly matche ViT's performance in large-scale settings. Besides, CRATE-α inherits the mathematical interpretability of the white-box models and can also achieve much better semantic interpretability evaluated by zero-shot segmentation.

A.3 Training details of CRATE-α-CLIPA models.

When employing the CRATE-α architecture to replace the vision encoder in the CLIPA [18] framework, we essentially follow the original CLIPA training recipe. The setup for the pre-training stage is presented in Table 9. During the fine-tuning stage, we made some modifications: the input image size is set to 224×224, the warmup steps are set to 800, and the base learning rate is set to 4e-7. When calculating the loss, we use the classification token from the vision encoder as the image feature and the last token from the text encoder as the text feature.

To explore the performance ceiling, we also train a ViT-CLIPA model from scratch. Most of the hyperparameters remain the same as those in Table 9, but there are some modifications in the pre-training stage. The batch size is set to 65,536, and the text length is set to 8 to speed up training. As

https://doi.org/10.52202/079017-1167

with the CLIPA setup, warm-up steps are set to 3,200. Additionally, we add color jitter and grayscale augmentation, and use global average pooling instead of the classification token. These modifications help stabilize training.

Config	Value
optimizer	AdamW [25]
optimizer momentum	(0.9, 0.95)
batch size	32768
base lr	8e-6
minimal lr	0
warm-up steps	1600
schedule	cosine decay [23]
weight decay	0.2
random crop area	(40, 100)
resize method	bi-linear
temperature init	1/0.07 [13, 19]

Table 9: **Pre-training hyper-parameters for CLIPA.**

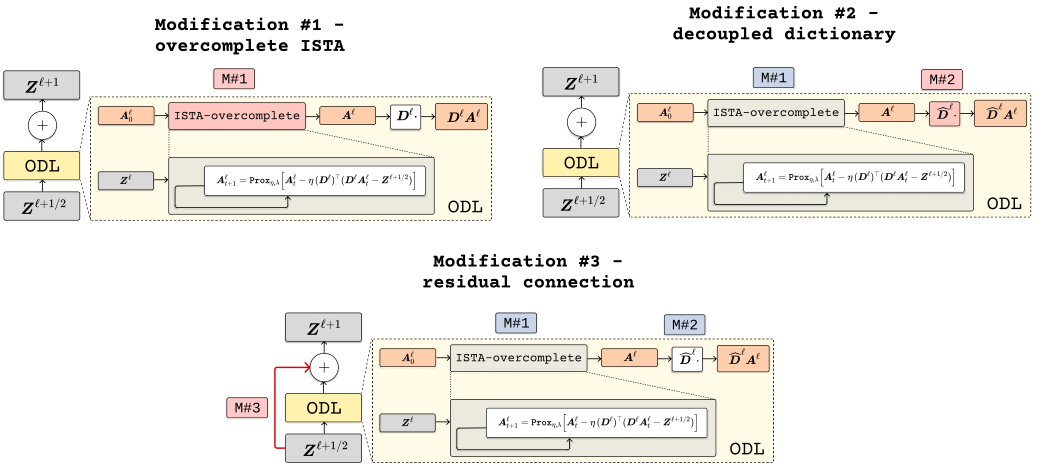

Figure 6: One layer of the CRATE-α model architecture (with more details for the three modifications described in Section 3.

Visualization of self-attention maps of CRATE-α. We provide visualization of attention maps of CRATE-α in Fig. 7.

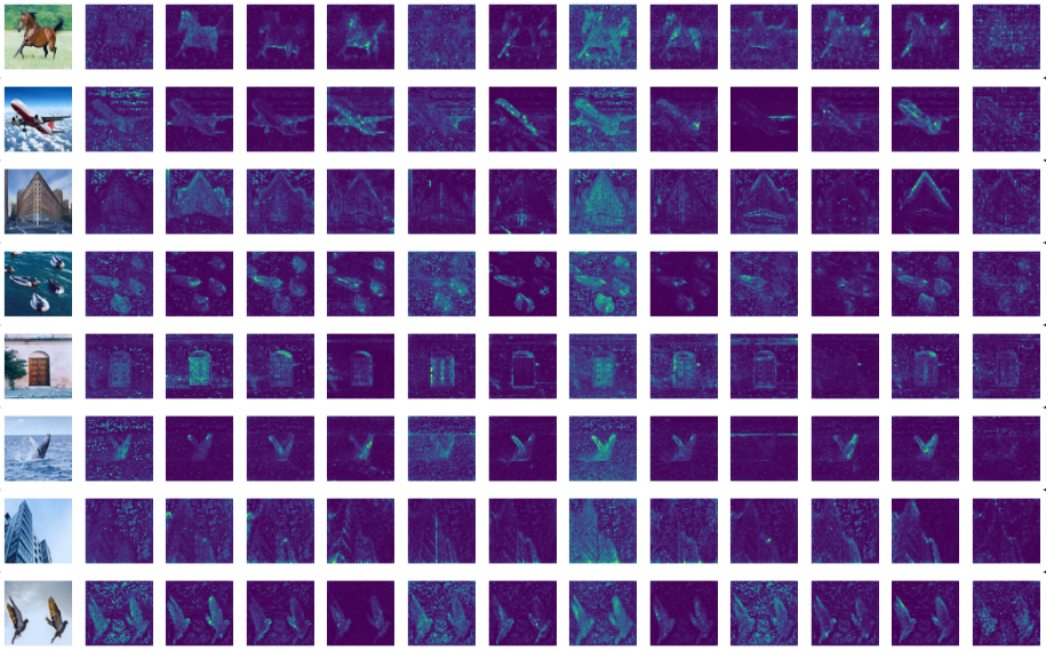

Figure 7: We visualize the self-attention maps of the CRATE-α Base model using 8×8 patches trained using classification. Similar to the original CRATE [47], our model also demonstrates the capability to automatically capture the structural information of objects. For each row, the original image is displayed on the left, while the corresponding self-attention maps are shown on the right. The number of self-attention maps corresponds to the number of heads in the CRATE-α model.

Visualization of loss curves. We visualize the training loss curves of the four models, including CRATE and its three variants, in Fig. 8. We visualize the training loss curves of CRATE-α-Base with different patch sizes in Fig. 9. In Fig. 10, we also visualize the training loss curves of models trained with efficient scaling strategy described in Section 4.4 in the main paper.

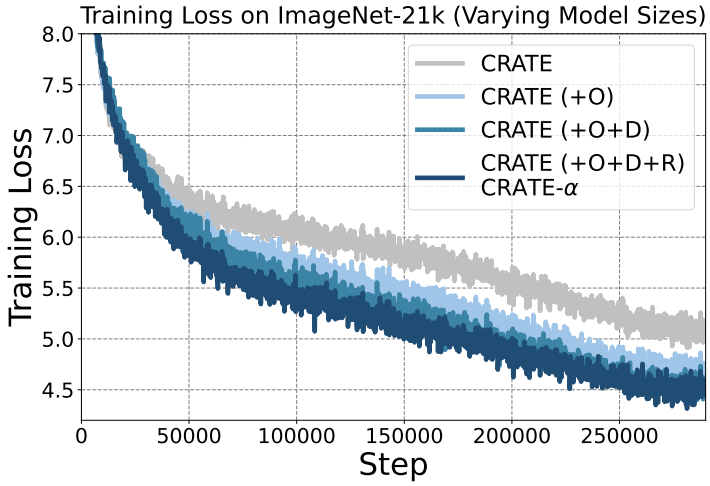

Figure 8: Training loss curves of different model architectures (mentioned in Fig. 1 in the main paper) on ImageNet-21K. The patch size is 32 for all four models shown in this figure. (+O: +overcomplete dictionary, +D: +decoupled dictionary, +R: +residual connection.)

https://doi.org/10.52202/079017-1167

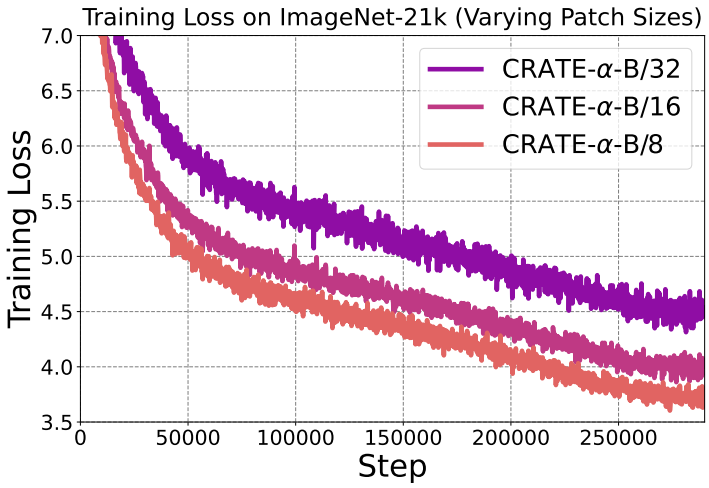

Figure 9: Comparing training loss curves across CRATE-α-Base with different patch sizes.

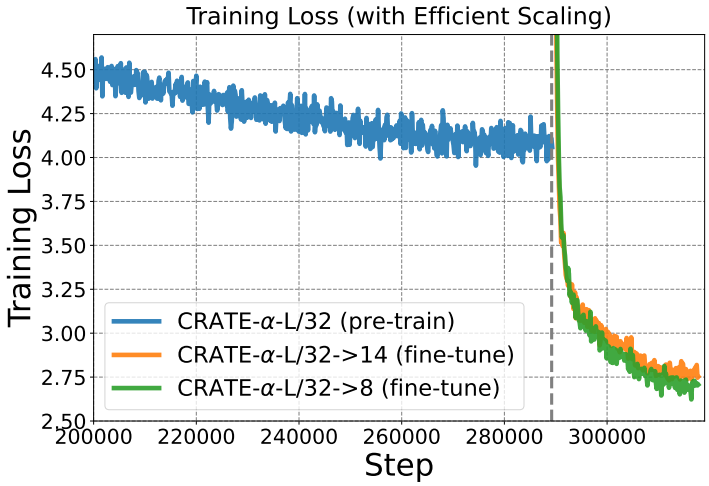

Figure 10: Comparing training loss curves when using the efficient scaling strategy. The blue curve corresponds to the CRATE-α-Large/32 model (in the pre-training stage). After pre-training the CRATE-α-Lage/32, we further fine-tune it with smaller patch sizes (longer token length), including patch size 14 (orange curve) and patch 8 (green curve).

https://doi.org/10.52202/079017-1167

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the primary contributions and scope of the paper. For specific details, refer to abstract and Sections 1, which align with the claims stated.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

https://doi.org/10.52202/079017-1167

Justification: We discuss the limitations of the work in section 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide details of reproducing the experimental results in Section 4 and the appendix.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We use public datasets and will also release the code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

https://doi.org/10.52202/079017-1167

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide details of reproducing the experimental results in Section 4 and the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We evaluate our method on a large public validation dataset, which is representative.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details about the training compute resources in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: we follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: we discuss them in Section 5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work has no potential risk for misuse.

https://doi.org/10.52202/079017-1167

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets used in the paper are properly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: we haven't introduced new assets in this work.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-1167