An In-depth Investigation of Sparse Rate Reduction in Transformer-like Models

Yunzhe Hu
School of Computing and Data Science
The University of Hong Kong
yzhu@cs.hku.hk

Difan Zou
School of Computing and Data Science
& Institute of Data Science
The University of Hong Kong
dzou@cs.hku.hk

Dong Xu
School of Computing and Data Science
The University of Hong Kong
dongxu@cs.hku.hk

Abstract

Deep neural networks have long been criticized for being black-box. To unveil the inner workings of modern neural architectures, a recent work [45] proposed an information-theoretic objective function called Sparse Rate Reduction (SRR) and interpreted its unrolled optimization as a Transformer-like model called Coding Rate Reduction Transformer (CRATE). However, the focus of the study was primarily on the basic implementation, and whether this objective is optimized in practice and its causal relationship to generalization remain elusive. Going beyond this study, we derive different implementations by analyzing layer-wise behaviors of CRATE, both theoretically and empirically. To reveal the predictive power of SRR on generalization, we collect a set of model variants induced by varied implementations and hyperparameters and evaluate SRR as a complexity measure based on its correlation with generalization. Surprisingly, we find out that SRR has a positive correlation coefficient and outperforms other baseline measures, such as path-norm and sharpness-based ones. Furthermore, we show that generalization can be improved using SRR as regularization on benchmark image classification datasets. We hope this paper can shed light on leveraging SRR to design principled models and study their generalization ability.

1 Introduction

Transformers [39, 11] have become the de facto choice of neural architecture nowadays and find great success in applications across language, vision, speech, and other scientific fields. The self-attention module in Transformers utilize global interactions to capture long-range dependency. However, the mechanisms and learning process of self-attention and other components in Transformers remain open problems, calling for more research to interpret and understand their properties.

One approach to interpreting the attention module involves experimental observation of the attention module to gain insights into their behaviors. For instance, DINO [7] provides a means to observe and analyze attention maps w.r.t class tokens in Vision Transformer (ViT), shedding light on their emerging interpretability from self-supervised learning. Another line of work focuses on interpreting or building attention module and even Transformer-like models from a mathematical perspective. Works in this vein have attempted to establish connections between Transformers and a reverse-engineered energy

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

function [43, 17], associative memory such as modern Hopfield network [35, 37, 26, 4] and sparse distributed memory [5], or programming languages [41, 22], to name a few.

Recently, the study of algorithm unrolling has emerged as a promising technique to bridge the gap between iterative optimization and neural architecture. A work by Yu et al. [45] considers the objective of representation learning as optimizing the Sparse Rate Reduction (SRR), a function that promotes maximum information gain described by the coding rate function [24, 46] and induces sparsity. In particular, they show that a Multi-head Subspace Self-Attention (MSSA) operator with skip connection and an Iterative Shrinkage-Thresholding Algorithms (ISTA) operator can be derived under some assumptions by unrolling minimization of the coding rate of representations in incoherent subspaces, i.e., compression and sparse coding, i.e., sparsification, respectively. By stacking these operations into layers, they build a Transformer-like model CRATE in which every layer should have the completely interpretable *compress-then-sparsify* behavior. However, although motivated by an information-theoretic and principled objective, it is still unexplored whether the core component MSSA operator with skip connection indeed implements the idea of compression in practice and how information propagates in the forward pass. On the other hand, SRR as the objective of representation learning is still an empirical formulation. Its causal relationship to generalization remains elusive.

In this paper, we conduct an in-depth investigation of this Transformer-like model and take steps to address these limitations. Our contributions are summarized as follows:

- In Section 4, we highlight the derivation artifacts through analysis of the key component MSSA operator and explore implementation variants of CRATE by inspecting the layer-wise behaviors. We show that the gradient approximation of the compression term will yield a counterproductive effect, performing *decompression* of token representations instead.
- In Section 5, we uncover the correlation between the learning objective SRR and generalization in unrolled models. By training models with varied hyperparameters, we show that SRR as a complexity measure has a positive correlation coefficient and outperforms other baselines.
- In Section 6, we demonstrate the effectiveness of SRR as a regularization technique for improved performance on benchmark datasets. Specifically, we show that the classification accuracy of unrolled models on CIFAR-10/100 can be consistently improved using a simple and efficient implementation of regularization.

2 Related Work

2.1 Interpreting Transformers

Research on interpreting Transformers [39, 11] has surged recently. Despite its achievements, the mechanisms and learning of attention layers remain enigmatic. One approach to interpreting Transformers is to experimentally observe the inner representations or output of key components like self-attention. This includes analysis by projecting parameters of Transformers to embedding space [10], inspecting the representations with another language model [14], visualizing attention map [7, 8, 44], etc. Several works opt for "mechanistic interpretability" [12, 28, 40] aiming to reverse-engineer the representations learned by Transformers that have "grokked" or mastered complex modular arithmetic task [34] and other synthetic tasks [23, 47]. Another line of work focuses more on theoretical understanding and building connections to other concepts. These papers utilize tools such as Bayesian inference [1], convex optimization [36] to analyze attention in Transformers. There have also been attempts to interpret a Transformer as an energy function optimizer [43, 17], connect attention to memory [35, 37, 26, 4, 5] or interacting particle systems [13] or transform into human-readable programs [41, 22], to name just a few. Our work focuses on the empirical investigation of a Transformer-like model, CRATE [45], recently introduced from pure mathematical derivation.

2.2 Algorithm Unrolling

Algorithm unrolling [27] has emerged as a promising technique for designing interpretable and efficient deep learning architectures. This approach establishes a direct connection between iterative algorithms and neural architecture, with each iteration of the algorithm corresponding to one layer of the architecture. Previous works have employed this technique to design popular networks in a forward-constructed manner. For instance, the seminal work [15] proposed to unroll the Iterative

Shrinkage-Thresholding Algorithm for sparse coding into layers of linear operation followed by ReLU non-linearity. Other works have tried to find a representation objective function to unroll into convolutional neural network [33, 6], graph neural network [42], and Transformers [43, 17]. We will follow this iteration-layer correspondence to conduct layer-wise analysis.

3 Revisiting Sparse Rate Reduction

Let $\boldsymbol{Z} = [\boldsymbol{z}_1, \dots, \boldsymbol{z}_N] \in \mathbb{R}^{d \times N}$ denote N samples, where each column $\boldsymbol{z}_i \in \mathbb{R}^d$ represents tokens in Transformers. $\boldsymbol{U} = [\boldsymbol{U}_1, \dots, \boldsymbol{U}_K] \in \mathbb{R}^{d \times Kp}$ denote a set of incoherent basis spanning K subspaces, wherein columns of $\boldsymbol{U}_i \in \mathbb{R}^{d \times p}$ represent basis in i-th low-dimensional subspace ($p < d$). We follow the configuration that $d = Kp$ as in standard ViT [11].

Previously, Yu et al. [46] propose that the compactness of representations $\boldsymbol{Z} \in \mathbb{R}^{d \times N}$ can be measured by a coding rate function: $R(\boldsymbol{Z}) \doteq \frac{1}{2} \log\det(\boldsymbol{I} + \frac{d}{N\epsilon^2} \boldsymbol{Z}^T \boldsymbol{Z})$. A more recent study [45] contends that the objective of representation learning is to transform and compress samples from an unknown distribution to a mixture of low-dimensional Gaussian distributions supported on incoherent bases. This objective boils down to the maximization of *Sparse Rate Reduction* (SRR):

$$\max_{\boldsymbol{Z} \in \mathbb{R}^{d \times N}} R(\boldsymbol{Z}) - R^c(\boldsymbol{Z}; \boldsymbol{U}) - \lambda \|\boldsymbol{Z}\|_0, \tag{1}$$

where $\|\cdot\|_0$ means ℓ_0 norm and $R^c(\boldsymbol{Z}; \boldsymbol{U}) \doteq \sum_{k=1}^K R(\boldsymbol{U}_k^T \boldsymbol{Z})$ measures the compactness of representations in the low-dimensional subspaces. One layer of a network, formulated as a mapping $f_{\boldsymbol{w}}(\cdot)$ parameterized by $\boldsymbol{w}$, can be interpreted as applying one step of gradient-based methods to the objective in (1). In practice, Yu et al. [45] use alternating minimization to break down the optimization into two steps: *compression*, i.e. $\min_{\boldsymbol{Z}} R^c(\boldsymbol{Z}; \boldsymbol{U})$ and *Sparsification*, i.e. $\min_{\boldsymbol{Z}} \lambda \|\boldsymbol{Z}\|_0 - R(\boldsymbol{Z})$. Specifically, given representation $\boldsymbol{Z}^{\ell-1}$ at $(\ell-1)$-th layer, $\boldsymbol{Z}^\ell$ can be obtained by two-step optimization:

$$\boldsymbol{Y}^\ell = \boldsymbol{Z}^{\ell-1} - \alpha \nabla R^c(\boldsymbol{Z}^{\ell-1}; \boldsymbol{U}^\ell) \approx \boldsymbol{Z}^{\ell-1} + \alpha\gamma^2 \,\mathrm{MSSA}(\boldsymbol{Z}^{\ell-1}; \boldsymbol{U}^\ell), \tag{2}$$

$$\boldsymbol{Z}^\ell = \mathrm{ReLU}\left(\boldsymbol{Y}^\ell + \beta(\boldsymbol{D}^\ell)^T(\boldsymbol{Y}^\ell - \boldsymbol{D}^\ell \boldsymbol{Y}^\ell) - \beta\lambda\boldsymbol{1}\right), \tag{3}$$

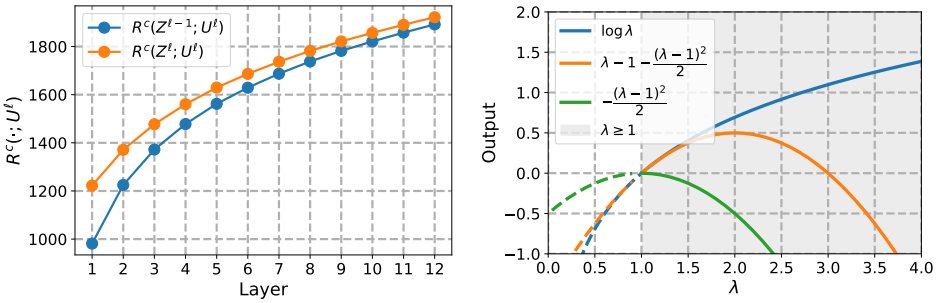
where $\alpha, \beta > 0$ are step sizes, $\boldsymbol{D}^\ell \in \mathbb{R}^{d \times d}$ is assumed as a complete dictionary, scalar $\gamma \doteq \frac{p}{N\epsilon^2}$ and

$$\begin{aligned} \mathrm{MSSA}(\boldsymbol{Z}; \boldsymbol{U}) &= \sum_{k=1}^K \boldsymbol{U}_k \boldsymbol{U}_k^T \boldsymbol{Z} \,\mathrm{softmax}((\boldsymbol{U}_k^T \boldsymbol{Z})^T (\boldsymbol{U}_k^T \boldsymbol{Z})) \\ &= [\boldsymbol{U}_1, \dots, \boldsymbol{U}_K] \begin{bmatrix} \boldsymbol{U}_1^T \boldsymbol{Z} \,\mathrm{softmax}((\boldsymbol{U}_1^T \boldsymbol{Z})^T (\boldsymbol{U}_1^T \boldsymbol{Z})) \\ \vdots \\ \boldsymbol{U}_K^T \boldsymbol{Z} \,\mathrm{softmax}((\boldsymbol{U}_K^T \boldsymbol{Z})^T (\boldsymbol{U}_K^T \boldsymbol{Z})) \end{bmatrix} \end{aligned} \tag{4}$$

The operator $\mathrm{MSSA}(\cdot; \boldsymbol{U})$ in (4), called the Multi-head Subspace Self-Attention (MSSA) operator, takes the form of self-attention in standard Transformers [39, 11], with tied query, key and value matrix, i.e., $\boldsymbol{U}_k^T$ while the output matrix being its transpose, i.e. $\boldsymbol{U}_k$. Instead of strictly following this formulation, they further replace $[\boldsymbol{U}_1, \dots, \boldsymbol{U}_K] \in \mathbb{R}^{d \times Kp}$ in the MSSA operator with an additional learnable parameter $\boldsymbol{W} \in \mathbb{R}^{d \times Kp}$. To distinguish them, we name the model with implementation (4) **CRATE-C(onceptual)**. By incrementally optimizing (1) with alternating minimization, a Transformer-like model with layered structures can be naturally constructed. With input $\boldsymbol{Z}^0$, e.g. tokenized images in ViT, an L-layer model iteratively optimizes the input and yields the final representations $\boldsymbol{Z}^L$. Parameters $\{\boldsymbol{U}^\ell\}_{\ell=1}^L$ and $\{\boldsymbol{D}^\ell\}_{\ell=1}^L$ can be learned through end-to-end training [15].

4 Is Sparse Rate Reduction Optimized in Transformer-like Models?

While the white-box Transformer-like model proposed in [45] is derived by unrolling optimization upon a pre-defined objective function, whether the optimization is implemented by the model in the forward pass is still unclear. In this section, we first review the main derivations at the core of building CRATE, i.e. unrolling optimization $\min_{\boldsymbol{Z}} R^c(\boldsymbol{Z}; \boldsymbol{U})$ into MSSA operator with skip connection as in (2), and identify the pitfalls in implementing the minimization. We then provide variant models based on different implementations and empirically show their layer-wise behaviors.



(a) Changes of coding rate projected on the same set of basis at each layer.

(b) Different approximation of $\log(\cdot)$.

Figure 1: In a simplified attention-only experiment, MSSA operator with skip connection actually implements an ascent method on $R^c(\boldsymbol{Z};\boldsymbol{U})$, opposed to its design purpose (*left*). This is due to an artifact in approximation with its second-order term. (*right*)

4.1 Pitfalls in Deriving CRATE-C

We first show that the second-order Taylor expansion of the coding rate of representations $\boldsymbol{Z}$ projected onto subspaces can be expressed as:

$$\begin{aligned} R^c(\boldsymbol{Z};\boldsymbol{U}) = \sum_{k=1}^{K}\sum_{i=1}^{N}\frac{1}{2}\log\lambda_i^k &\geq \sum_{k=1}^{K}\sum_{i=1}^{N}\frac{1}{2}\left(\lambda_i^k - 1 - \frac{(\lambda_i^k-1)^2}{2}\right) \\ &= \sum_{k=1}^{K}\left(\underbrace{\frac{\gamma}{2}\|\boldsymbol{U}_k^T\boldsymbol{Z}\|_F^2}_{\text{First-order term}} \underbrace{-\frac{\gamma^2}{4}\|(\boldsymbol{U}_k^T\boldsymbol{Z})^T\boldsymbol{U}_k^T\boldsymbol{Z}\|_F^2}_{\text{Second-order term}}\right), \end{aligned} \tag{5}$$

where $\lambda_i^k \geq 1, i \in [N]$ are the eigenvalues of $\boldsymbol{I} + \gamma(\boldsymbol{U}_k^T\boldsymbol{Z})^T\boldsymbol{U}_k^T\boldsymbol{Z}$. Following the derivation and implementation from Appendix A.2 in [45], the MSSA operator with skip connection is constructed by performing an *approximation* of gradient descent on $R^c(\boldsymbol{Z};\boldsymbol{U})$:

$$\boldsymbol{Z} - \alpha\nabla_{\boldsymbol{Z}}R^c(\boldsymbol{Z};\boldsymbol{U}) = \boldsymbol{Z} - \alpha\gamma\sum_{k=1}^{K}\boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{Z}\left(\boldsymbol{I} + \gamma(\boldsymbol{U}_k^T\boldsymbol{Z})^T(\boldsymbol{U}_k^T\boldsymbol{Z})\right)^{-1} \tag{6}$$

$$\approx \boldsymbol{Z} - \alpha\left(\underbrace{\gamma\sum_{k=1}^{K}\boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{Z}}_{\nabla\text{of first-order term}} \underbrace{-\gamma^2\sum_{k=1}^{K}\boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{Z}(\boldsymbol{U}_k^T\boldsymbol{Z})^T(\boldsymbol{U}_k^T\boldsymbol{Z})}_{\nabla\text{of second-order term}}\right) \tag{7}$$

$$\approx \boldsymbol{Z} + \alpha\gamma^2\sum_{k=1}^{K}\boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{Z}\,\mathrm{softmax}((\boldsymbol{U}_k^T\boldsymbol{Z})^T(\boldsymbol{U}_k^T\boldsymbol{Z})). \tag{8}$$

It can be seen that this update step takes the gradient of a lower bound of $R^c(\boldsymbol{Z};\boldsymbol{U})$ and discards the first-order term. With a proper step size, the coding rate on the same subspaces is expected to decrease after one iteration. However, we will show that this is not the actual case via a toy experiment.

We consider a simplified setting where L layers of update (8) are conducted with parameters $\{\boldsymbol{U}^\ell\}_{\ell=1}^L$ initialized as orthonormal matrices. We initialize a random variable $\boldsymbol{Z}^0$ from a Gaussian distribution and measure the coding rate before and after each layer. We set $N = 196$, $L = 12$, $d = 384$, $K = 6$, $\alpha = 1$, and a proper ϵ^2 such that $\gamma = 1$. As shown in Figure 1a, $R^c(\boldsymbol{Z}^\ell;\boldsymbol{U}^\ell)$ is always greater than $R^c(\boldsymbol{Z}^{\ell-1};\boldsymbol{U}^\ell)$ and $R^c(\boldsymbol{Z}^\ell;\boldsymbol{U}^\ell)$ is increasing in general as the layer goes deeper. This means the update (8) that resembles the standard self-attention with skip connection does not essentially implement a descent method on R^c. The crux lies in the approximation of R^c's gradient.

When taking the gradient of R^c to construct the MSSA operator, omitting its first-order term will produce a counterproductive effect. As shown on the left-hand side of the inequality in (5), R^c can be

expressed as the sum of logarithms of eigenvalues. We expect the eigenvalues to decrease to minimize the value of R^c. Figure 1b illustrates different approximations of the logarithm function. If we omit the first-order term of its Taylor expansion and only perform descent methods on its second-order term (corresponding to $-\frac{(\lambda_i^k-1)^2}{2}$), the eigenvalues will go up leading to an increase in the value of R^c. Therefore, one step of update (8) secretly maximizes R^c, contrary to the purpose of its design. More figures detailing this issue are in Appendix A.

4.2 Producing CRATE Variants

In the previous subsection, we show the problems arising from gradient approximation when unrolling $\min_{\boldsymbol{Z}} R^c(\boldsymbol{Z};\boldsymbol{U})$ into MSSA operator with shortcut. We will, in the subsection, introduce two variants of CRATE induced by the conceptual and implementation gaps. These variants can be considered as the alternative instantiations of the optimization-induced architectures but in a more self-contained way. They also serve as representative samples for our subsequent investigations of SRR.

One variant of CRATE, motivated by the theoretical gap between CRATE-C and the SRR principle, could naturally emerge when the sign before the MSSA operator in (8) is changed. Similar to previous analysis via eigenvalues, this update of representations in fact implements one step of ascent methods on the second-order term of $\boldsymbol{R}^c$, therefore minimizing the eigenvalues and consequently $\boldsymbol{R}^c$. This variant is designed to counter the pitfalls in CRATE-C, enabling a more faithful reduction in $\boldsymbol{R}^c$ and thereby enhancing alignment with the SRR principle. We term the Transformer-like model with this implementation **CRATE-N(egative)**:

$$\boldsymbol{Z} - \alpha\gamma^2 \sum_{k=1}^{K} \boldsymbol{U}_k \boldsymbol{U}_k^T \boldsymbol{Z} \operatorname{softmax}((\boldsymbol{U}_k^T \boldsymbol{Z})^T (\boldsymbol{U}_k^T \boldsymbol{Z})). \tag{9}$$

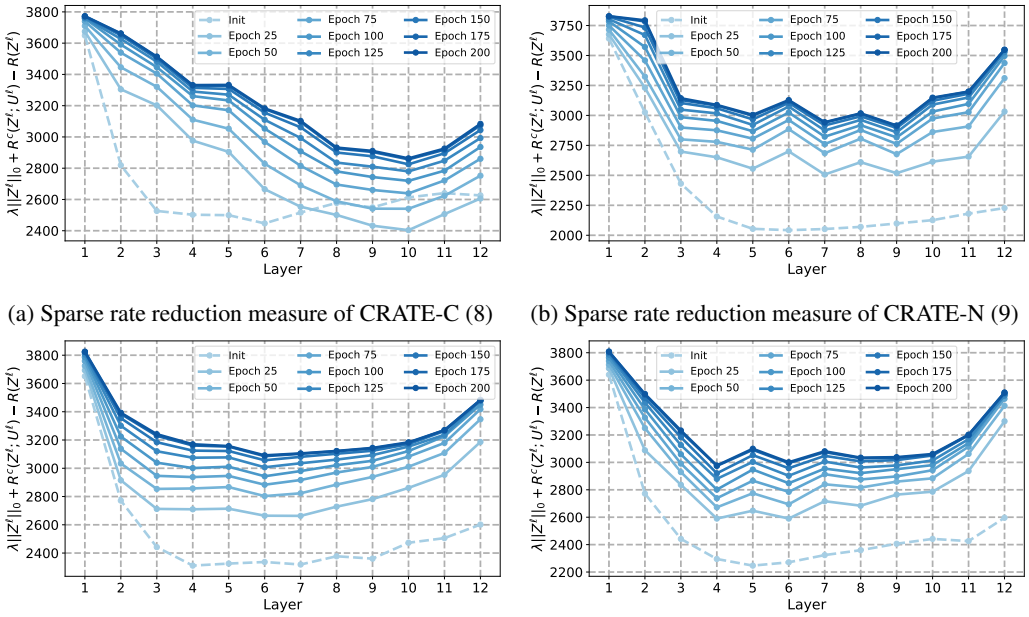
The other variant we would like to introduce is motivated by the misalignment between CRATE and CRATE-C. Although replacing the output matrix $[\boldsymbol{U}_1, \ldots, \boldsymbol{U}_K]$ with learnable parameters $\boldsymbol{W}$ in CRATE empirically boosts performance, it also contaminates the framework and sacrifices the mathematical interpretability. Does this modification really matter? Can we preserve model performance while maintaining framework integrity? It turns out that a simple transpose operation of the output matrix could greatly close the empirical gap to CRATE, without more parameters. Other manipulations and discussions can be found in Appendix B. We refer to the model with this simple manipulation **CRATE-T(ranspose)**:

$$\boldsymbol{Z} + \alpha\gamma^2 \left[\boldsymbol{U}_1, \ldots, \boldsymbol{U}_K\right]^T \begin{bmatrix} \boldsymbol{U}_1^T \boldsymbol{Z} \operatorname{softmax}((\boldsymbol{U}_1^T \boldsymbol{Z})^T (\boldsymbol{U}_1^T \boldsymbol{Z})) \\ \vdots \\ \boldsymbol{U}_K^T \boldsymbol{Z} \operatorname{softmax}((\boldsymbol{U}_K^T \boldsymbol{Z})^T (\boldsymbol{U}_K^T \boldsymbol{Z})) \end{bmatrix}. \tag{10}$$

4.3 Behaviors of Sparse Rate Reduction

The Transformer-like model CRATE is built by sequentially stacking the layer that comprises two modules in (2) and (3) (or different implementations). Although each module is designed to implement one-step optimization of different objectives, it is unclear whether the architecture design achieves the optimization as a whole. On the other hand, there is also a need to determine whether the model parameters learned through end-to-end training actually lead to improved optimization.

To investigate how sparse rate reduction evolves in the forward pass and during training, we train CRATE and its variants on CIFAR-10/100 datasets and evaluate the sparse rate reduction measure $\lambda\|\boldsymbol{Z}^\ell\|_0 + R^c(\boldsymbol{Z}^\ell;\boldsymbol{U}^\ell) - R(\boldsymbol{Z}^\ell)$ at different layers and epochs on the training set. λ is chosen as 0.1 and detailed experiment settings can be found in Section 6.1. Figure 2 and Figure 3 show the behaviors of sparse rate reduction of CRATE along with its variants CRATE-C, CRATE-N, and CRATE-T under Tiny configurations in [45]. When the models are randomly initialized, the sparse rate reduction measure almost monotonically decreases in the first 9 layers and then rises in the subsequent layers. This partly confirms the layer-wise optimization of the objective SRR and its alignment with forward architecture design, although in Section 4.1 we demonstrate that $R^c(\boldsymbol{Z};\boldsymbol{U})$ will monotonically go up in the absence of operation (3). We conjecture that the ReLU non-linearity may also play an important role in optimizing the compression term $R^c(\boldsymbol{Z};\boldsymbol{U})$ in the forward pass.



(a) Sparse rate reduction measure of CRATE-C (8)

(b) Sparse rate reduction measure of CRATE-N (9)

(c) Sparse rate reduction measure of CRATE-T (10)

(d) Sparse rate reduction measure of CRATE

Figure 2: Sparse rate reduction measure $\lambda\|\boldsymbol{Z}\|_0 + R^c(\boldsymbol{Z};\boldsymbol{U}) - R(\boldsymbol{Z})$ of CRATE and its variants evaluated at different layers and epochs on CIFAR-10.

Another surprising finding is that as the learning process proceeds, the sparse rate reduction measure at each layer will increase monotonically across all models, with a rare exception in the last few layers of CRATE-C.

These phenomena give us implications for understanding Transformer-like models: the representations of initialized models converge fast in the first few layers and hover around the local minimum of the objective landscape; however, the useful information in representations may be discarded due to over-compression and the learning of parameters gradually increases sparse rate reduction measure to counteract this effect for improved task-specific representations.

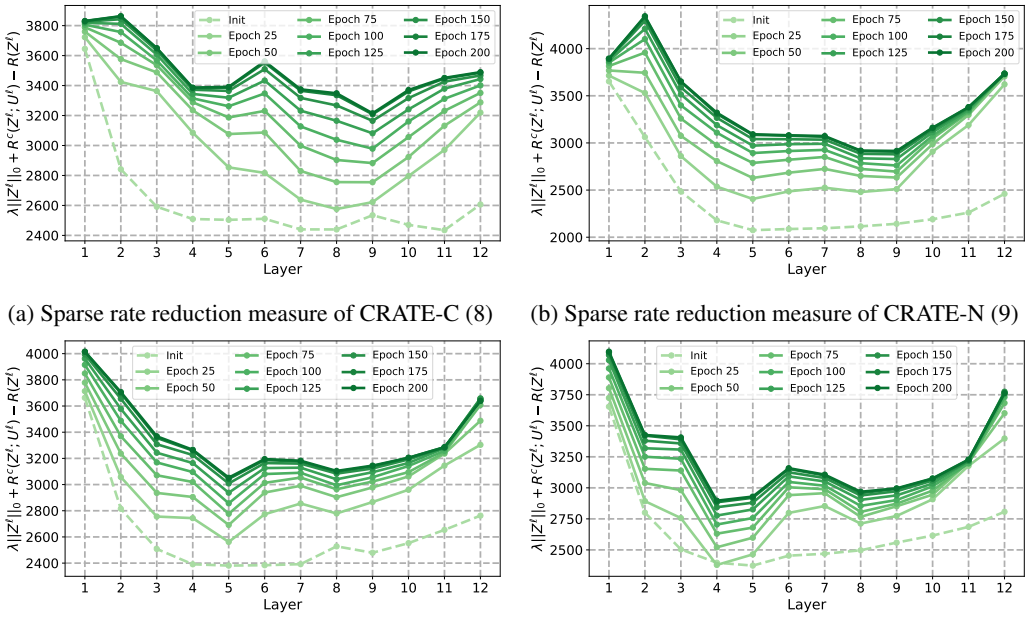
To summarize, our finding is that sparse rate reduction measure is incrementally optimized in a realistic setting at initialization. This aligns well with its design purpose from a macro perspective. With varied implementations, the result still holds even when the compression-inspired operator MSSA diverges from its goal from a micro perspective. We postulate that ReLU non-linearity in (3) could also promote compression and leave their interaction for future work.

5 Whether Sparse Rate Reduction Benefits Generalization?

So far, we have partially confirmed the validity of different implementations of Transformer-like models by inspecting the layer-wise optimization of SRR. But whether this objective is important or principled for these architectures to generalize is still an unaddressed problem. In this section, we want to explore the predictive power of SRR and its causal relationship to the generalization of CRATE.

5.1 Sparse Rate Reduction as a Complexity Measure

An important tool to study the generalization of deep networks is *complexity measure*. A complexity measure that can properly reflect the generalization needs to have the following property: lower complexity should indicate a smaller generalization gap. Complexity measures can be either theoretically motivated, such as PAC-Bayes [25, 29], VC-dimension [38], norm-based bounds [32, 3, 30] or empirically motivated, such as sharpness [20] and path-norm [31]. We choose to adapt SRR into a



(a) Sparse rate reduction measure of CRATE-C (8)

(b) Sparse rate reduction measure of CRATE-N (9)

(c) Sparse rate reduction measure of CRATE-T (10)

(d) Sparse rate reduction measure of CRATE

Figure 3: Sparse rate reduction measure $\lambda\|\boldsymbol{Z}\|_0 + R^c(\boldsymbol{Z};\boldsymbol{U}) - R(\boldsymbol{Z})$ of CRATE and its variants evaluated at different layers and epochs on CIFAR-100.

complexity measure that belongs to the latter category:

$$\mu_{\text{SRR}}(\boldsymbol{w};\boldsymbol{Z}) = \frac{1}{L}\sum_{\ell=1}^{L}\mu_{\text{SRR}}^{\ell}(\boldsymbol{w}^\ell;\boldsymbol{Z}^\ell) = \frac{1}{L}\sum_{\ell=1}^{L}\Big(\lambda\|\boldsymbol{Z}^\ell\|_0 + R^c(\boldsymbol{Z}^\ell;\boldsymbol{U}^\ell) - R(\boldsymbol{Z}^\ell)\Big), \tag{11}$$

where $\boldsymbol{Z}^\ell$ denotes the output at layer ℓ and $\boldsymbol{w}^\ell$ contains the parameters including $\boldsymbol{U}^\ell$ and $\boldsymbol{D}^\ell$.

5.2 Correlation with Generalization

An effective measure of complexity should bound the generalization gap, defined as the difference between validation loss and training loss when the latter reaches a threshold, i.e., $\mathcal{L}_{val} - \mathcal{L}_{train}$, with high probability. However, for those measures that do not provably bound this gap, as is the case with SRR measure (11), we need to evaluate its correlation with the generalization gap to understand its causal relationship to generalization.

Collecting Trained Models To evaluate the complexity measure and generalization across models, we consider changing the hyperparameters and collect a set of models trained to meet a specific stopping criterion. Here, we also consider the model type containing different variants of CRATE as a hyperparameter to investigate its influence on generalization. Formally, let Θ_i denote a type of hyperparameter with $|\Theta_i|$ different choices, and define $\boldsymbol{\theta} \doteq (\theta_1, \theta_2, \ldots, \theta_n) \in \Theta_1 \times \cdots \times \Theta_n$ as an instantiation from n types of hyperparameters. By varying choices across hyperparameter space, we can produce $|\Theta_1| \times \cdots \times |\Theta_n|$ models. In our experiment, we consider $n = 5$ hyperparameters, including *batch size*, *initial learning rate*, *width*, *dropout*, and *model type*. Each contains 2 choices except that the model type contains 4 implementations we discussed before. We successfully train a total of 64 models on CIFAR-10 dataset, when cross-entropy loss reaches 0.01 following the stopping criterion in [18]. Experimental details and choices of hyperparameters can be found in Appendix C.

Evaluation Criterion A common method for measuring correlation is by utilizing Kendall's rank correlation coefficient [19, 18], which ranges from -1 to 1. Generally, the closer the coefficient is to one, the stronger the causal relationship and the greater the predictive power a measure can offer for generalization. Zero value usually means independent relationships. For a given complexity

Table 1: Correlation of complexity measures with generalization gap (width $d = 384$).

Complexity measures	Batch size	Learning rate	Dropout	Model type	Overall τ	Ψ
ℓ_2-norm	0.200	-0.333	-0.333	-0.429	-0.363	-0.224
ℓ_2-norm-init	0.200	-0.200	-0.333	-0.286	-0.290	-0.158
# params	0.000	0.000	0.000	-0.572	-0.351	-0.143
1/margin	-0.067	0.467	0.467	0.238	0.415	0.276
sum-of-spec	0.200	-0.333	-0.467	-0.381	-0.290	-0.245
prod-of-spec	0.200	-0.333	-0.467	-0.476	-0.338	-0.269
sum-of-spec/margin	0.333	-0.333	-0.467	-0.048	-0.230	-0.129
prod-of-spec/margin	0.333	-0.333	-0.467	-0.143	-0.260	-0.152
fro/spec	-0.200	0.333	0.467	-0.476	0.019	0.031
spec-init-main	0.333	-0.333	-0.467	-0.190	-0.273	-0.164
spec-orig-main	0.200	-0.333	-0.467	-0.095	-0.252	-0.174
sum-of-fro	0.200	-0.333	-0.333	-0.381	-0.325	-0.212
prod-of-fro	0.200	-0.333	-0.333	-0.429	-0.372	-0.224
sum-of-fro/margin	0.333	-0.200	-0.467	-0.048	-0.217	-0.095
prod-of-fro/margin	0.333	-0.200	-0.467	-0.143	-0.247	-0.119
fro-distance	0.200	-0.200	-0.333	-0.286	-0.290	-0.155
spec-distance	0.200	-0.200	-0.333	-0.286	-0.290	-0.155
param-norm	0.200	-0.333	-0.333	-0.429	-0.363	-0.224
path-norm	0.333	-0.600	-0.467	-0.286	-0.191	-0.255
pac-bayes-init	0.200	0.200	-0.600	0.238	0.015	-0.009
pac-bayes-orig	-0.200	0.333	0.467	0.381	0.333	0.245
$1/\sigma$ pac-bayes-flatness	-0.267	0.333	0.333	0.455	0.333	0.213
SRR	-0.067	0.467	0.333	0.714	**0.445**	**0.362**

measure, we can construct a set of samples $\mathcal{T}$ containing the measure $\mu(\boldsymbol{\theta})$ and generalization gap $g(\boldsymbol{\theta})$ evaluated at different combinations of hyperparameters $\boldsymbol{\theta}$ and calculated Kendall's coefficient on this set:

$$\mathcal{T} \triangleq \cup_{\boldsymbol{\theta} \in \Theta_1 \times \cdots \times \Theta_n} \{(\mu(\boldsymbol{\theta}), g(\boldsymbol{\theta}))\}, \tag{12}$$

$$\tau(\mathcal{T}) \triangleq \frac{1}{|\mathcal{T}|(|\mathcal{T}|-1)} \sum_{(\mu_1, g_1) \in \mathcal{T}} \sum_{(\mu_2, g_2) \in \mathcal{T} \setminus (\mu_1, g_1)} \operatorname{sign}(\mu_1 - \mu_2) \operatorname{sign}(g_1 - g_2). \tag{13}$$

Experimental Results. In our experiment, we find that the correlation of various measures with the generation can be reflected with more prominence under a selected width. Accordingly, we present the results in terms of Kendall's coefficient τ in Table 1 and scatter plot of SRR measure in Figure 4 when the width d is chosen as 384. The results when $d = 768$ is deferred to Appendix D. The granulated coefficient Ψ is also reported (see [18] for a detailed definition).

We confirm the findings from prior works that some norm-based measures, such as sum/prod of spectral/Frobenius norm of parameters negatively correlate with generalization, even on Transformer-like models. An interesting finding is that path-norm also negatively correlates with generalization, which partly contradicts the previous conclusion. This implies that regularization on path-norm, e.g. Path-SGD [31], may not be applicable for improved generalization on Transformer-like models. Among the measures we investigated, the inverse of margin and sharpness-based PAC-Bayes flatness show positive and strong correlations. This result justifies the common belief that larger margin or flatter loss landscape leads to better generalization across the investigated Transformer-like models. Compared to baselines, the SRR measure in (11)

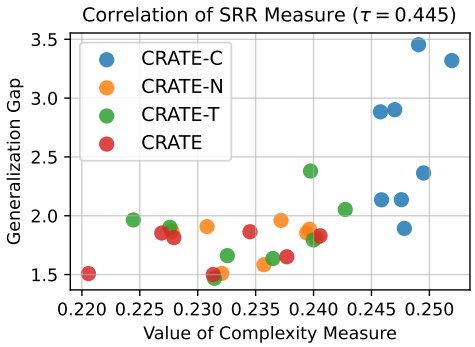


Figure 4: A scatter plot illustrating the value of SRR measure and generalization gap across CRATE and variants with network width $d = 384$.

achieves the highest overall coefficient and, particularly in the model type axis, outperforms the rest. This motivates the use of SRR as regularization in the loss function to improve generalization.

6 Sparse Rate Reduction as Regularization

Since SRR measure enjoys a strong correlation to the generalization of Transformer-like models, we would like to investigate its potential as the direct regularization to the standard training loss. In particular, we add the SRR measure in (11) by a regularization factor η to the cross-entropy loss:

$$\min_{\boldsymbol{w}} \mathcal{L}_{\text{ce}}(\boldsymbol{w}) + \lambda \cdot \frac{1}{L} \sum_{\ell=1}^{L} \mu^{\ell}_{\text{SRR}}(\boldsymbol{w}^{\ell}; \boldsymbol{Z}^{\ell}_{\text{StopGrad}}), \tag{14}$$

where $\lambda > 0$ is the regularization coefficient and $\boldsymbol{Z}^{\ell}_{\text{StopGrad}} = f_{\boldsymbol{w}^{\ell}}(\text{StopGrad}(\boldsymbol{Z}^{\ell-1}))$. The operator StopGrad here, implemented as "$\text{Tensor.detach}()$" in PyTorch, prevents gradient propagation from the output $\boldsymbol{Z}^{\ell}$ to the previous layers. This allows parameters $\boldsymbol{w}^{\ell}$ at each layer to be updated without interfering with each other, giving more precise optimization of SRR in separate layers.

6.1 Experiment Settings

Model Configurations We follow the configuration of CRATE-Tiny in [45] in this experiment. Specifically, we set the depth $L = 12$, width $d = 384$, number of subspaces $K = 6$, step size $\alpha = 1$, and scaling factor $\gamma = 1$. We also include LayerNorm before each operation in (2)(3) for better trainability and learnable positional encoding. A trainable [CLS] token is prepended to the representations for computing cross-entropy loss and classification.

Datasets and Optimization We use CIFAR-10 and CIFAR-100 datasets for training and evaluation. In practice, we adopt Adam [21] optimizer and initialize learning rate as 1×10^{-4} with cosine decay. All models are trained for 200 epochs with batch size as 128. Note that we only use the basic data augmentations: random resize and cropping, horizontal flipping, and RandAugment [9] (with the number transformations $n = 2$ and magnitude $m = 14$). We do not use other techniques for state-of-the-art performance but to demonstrate the effectiveness of SRR as regularization. We tune the factor η via a grid search over $\{0.0001, 0.001, 0.01, 0.1, 1\}$ and find that 0.001 works best. All experiments are conducted on NVIDIA GeForce RTX 3090.

Table 2: Top-1 accuracy for CRATE and its variants trained with or without SRR regularization on CIFAR-10/100 from scratch (width $d = 384$).

Models	CIFAR-10		CIFAR-100	
	cross-entropy	+ SRR regularization (L=12)	cross-entropy	+ SRR regularization (L=12)
CRATE-C	76.87	**77.61**	43.40	**44.53**
CRATE-N	81.52	**81.91**	55.11	**55.62**
CRATE-T	85.49	**85.52**	60.59	**60.69**
CRATE	86.67	**86.79**	62.40	**62.52**

6.2 Efficient Implementation

Regularizing the training loss with sparse rate reduction measure (11) needs to compute $R(\boldsymbol{Z})$ and $R^c(\boldsymbol{Z}; \boldsymbol{U})$ for every layer. However, this is highly inefficient as it involves high-dimensional matrix multiplication, and it lacks flexibility in controlling parameters. To alleviate this issue, we implement efficient regularization as per layer regularization or random layer regularization: select a pre-defined layer or a random layer with uniform probability during training. In practice, we find that the former works better. Table 2 provides the results of CRATE and its variants trained from scratch on CIFAR-10/100. SRR regularization is sufficient to improve the performance by simply leveraging the last layer. We also provide a comparison of efficient implementations in Appendix E

7 Conclusion

To further research in interpreting neural architecture, we provide an in-depth investigation of a recent mathematically driven Transformer-like model, CRATE. Although designed with a principled objective, we identify an artifact in its forward construction and show that the simplest implementation can have the opposite effect in realizing its designed goal. We then provide implementation variants and investigate their layer-wise behaviors in optimizing SRR. An interesting finding is that alternative models exhibit similar behaviors, validating the use of SRR in designing Transformer-like models. Furthermore, we demonstrate its positive correlation to generalization and effectiveness over baselines. Driven by this connection, we show a simple way to use SRR as regularization to improve performance on CIFAR-10/100 datasets. Future direction may include applying layer-wise training and connecting SRR to the Forward-Forward algorithm [16], or exploring the impact of depth in the unrolled models.

Limitations

This study has several limitations. Firstly, the conclusion that the SRR measure can be a strong indicator of generalization is limited to the CRATE family. Generalizing this conclusion to standard Transformers would be non-trivial, as the SRR measure is not properly defined when the query-key-value matrices have independent learnable parameters instead of shared ones. Secondly, the performance of a more faithful implementation (CRATE-N) falls behind the one with a simple manipulation (CRATE-T). This calls for a rigorous inspection of each component's functionality in the framework. Lastly, while we confirm the positive correlation to generalization, our analysis is limited in scale. Consequently, drawing definitive conclusions regarding whether SRR can be a principle or necessitates further engineering to push the model's limit is challenging. A Better and more systematic way is needed to determine whether SRR is principled for designing the Transformer-like models and quantify this relationship in an appropriate task, perhaps beyond classification.

Acknowledgments

This work was supported in part by Natural Science Fund China (62306252), in part by the Hong Kong Research Grants Council General Research Fund (17203023), in part by the Hong Kong Research Grants Council Collaborative Research Fund (C5052-23G), in part by The Hong Kong Jockey Club Charities Trust under Grant 2022-0174, in part by the Startup Fund and the Seed Fund for Basic Research for New Staff from The University of Hong Kong, and in part by the funding from UBTECH Robotics.

References

[1] Bang An, Jie Lyu, Zhenyi Wang, Chunyuan Li, Changwei Hu, Fei Tan, Ruiyi Zhang, Yifan Hu, and Changyou Chen. Repulsive attention: Rethinking multi-head attention as bayesian inference. *arXiv preprint arXiv:2009.09364*, 2020.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

[4] Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint. *Advances in Neural Information Processing Systems*, 36, 2023.

[5] Trenton Bricken and Cengiz Pehlevan. Attention approximates sparse distributed memory. *Advances in Neural Information Processing Systems*, 34:15301–15315, 2021.

[6] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

[7] 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*, pages 9650–9660, 2021.

[8] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 782–791, 2021.

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[10] Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535*, 2022.

[11] 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. In *International Conference on Learning Representations*, 2020.

[12] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[13] Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. *Advances in Neural Information Processing Systems*, 36, 2023.

[14] Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscope: A unifying framework for inspecting hidden representations of language models. *arXiv preprint arXiv:2401.06102*, 2024.

[15] 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.

[16] Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2022.

[17] Benjamin Hoover, Yuchen Liang, Bao Pham, Rameswar Panda, Hendrik Strobelt, Duen Horng Chau, Mohammed Zaki, and Dmitry Krotov. Energy transformer. *Advances in Neural Information Processing Systems*, 36, 2023.

[18] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

[19] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[20] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] David Lindner, János Kramár, Sebastian Farquhar, Matthew Rahtz, Tom McGrath, and Vladimir Mikulik. Tracr: Compiled transformers as a laboratory for interpretability. *Advances in Neural Information Processing Systems*, 36, 2023.

[23] Ziming Liu, Eric J Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *The Eleventh International Conference on Learning Representations*, 2022.

[24] Yi Ma, Harm Derksen, Wei Hong, and John Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE transactions on pattern analysis and machine intelligence*, 29(9):1546–1562, 2007.

[25] David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170, 1999.

[26] Beren Millidge, Tommaso Salvatori, Yuhang Song, Thomas Lukasiewicz, and Rafal Bogacz. Universal hopfield networks: A general framework for single-shot associative memory models. In *International Conference on Machine Learning*, pages 15561–15583. PMLR, 2022.

[27] Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021.

[28] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

[29] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

[30] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

[31] Behnam Neyshabur, Russ R Salakhutdinov, and Nati Srebro. Path-sgd: Path-normalized optimization in deep neural networks. *Advances in neural information processing systems*, 28, 2015.

[32] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on learning theory*, pages 1376–1401. PMLR, 2015.

[33] Vardan Papyan, Yaniv Romano, Jeremias Sulam, and Michael Elad. Theoretical foundations of deep learning via sparse representations: A multilayer sparse model and its connection to convolutional neural networks. *IEEE Signal Processing Magazine*, 35(4):72–89, 2018.

[34] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

[35] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

[36] Arda Sahiner, Tolga Ergen, Batu Ozturkler, John Pauly, Morteza Mardani, and Mert Pilanci. Unraveling attention via convex duality: Analysis and interpretations of vision transformers. In *International Conference on Machine Learning*, pages 19050–19088. PMLR, 2022.

[37] Danil Tyulmankov, Ching Fang, Annapurna Vadaparty, and Guangyu Robert Yang. Biological learning in key-value memory networks. *Advances in Neural Information Processing Systems*, 34:22247–22258, 2021.

[38] Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity: festschrift for alexey chervonenkis*, pages 11–30. Springer, 2015.

[39] 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.

[40] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

[41] Gail Weiss, Yoav Goldberg, and Eran Yahav. Thinking like transformers. In *International Conference on Machine Learning*, pages 11080–11090. PMLR, 2021.

[42] Yongyi Yang, Tang Liu, Yangkun Wang, Jinjing Zhou, Quan Gan, Zhewei Wei, Zheng Zhang, Zengfeng Huang, and David Wipf. Graph neural networks inspired by classical iterative algorithms. In *International Conference on Machine Learning*, pages 11773–11783. PMLR, 2021.

[43] Yongyi Yang, David P Wipf, et al. Transformers from an optimization perspective. *Advances in Neural Information Processing Systems*, 35:36958–36971, 2022.

[44] Catherine Yeh, Yida Chen, Aoyu Wu, Cynthia Chen, Fernanda Viégas, and Martin Wattenberg. Attentionviz: A global view of transformer attention. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[45] 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.

[46] Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *Advances in Neural Information Processing Systems*, 33:9422–9434, 2020.

[47] Yi Zhang, Arturs Backurs, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, and Tal Wagner. Unveiling transformers with lego: a synthetic reasoning task. *arXiv preprint arXiv:2206.04301*, 2022.

A Complete Demonstrations of the Pitfalls

To give a clearer picture of how approximations affect the optimization of R^c, we provide the complete results with different update rules under the same simplified settings in the main text:

(a) Gradient descent on R^c.

(b) Gradient descent on the second-order Taylor expansion of R^c.

(c) Gradient descent on the first-order term of the Taylor expansion of R^c (w/o second-order term).

(d) Gradient descent on the second-order term of the Taylor expansion of R^c (w/o first-order term).

(e) Further adding softmax function upon (d).

The results in Figure 5 correspond to the above experiments. Gradient descent on R^c did make it decrease across layers. Conversely, applying gradient descent on its second-order Taylor expansion resulted in an increase, indicating a potentially flawed approximation. Isolating gradient descent to the second-order term led to a rise in R^c, as opposed to the design purpose. Furthermore, incorporating the softmax function, a real-world operation examined in the main text, did not alter this conclusion.

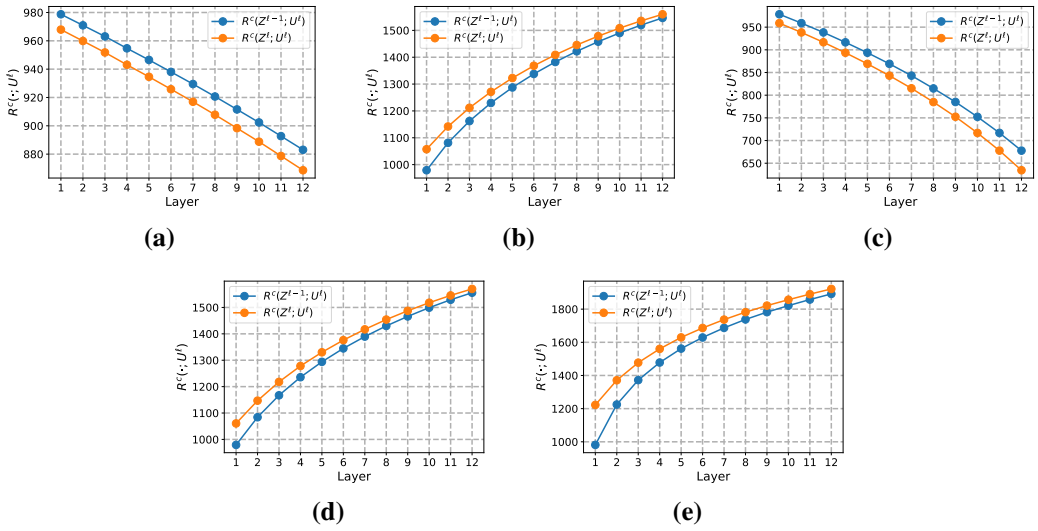


Figure 5: **(a)** Original gradient update, i.e, (6) **(b)** update from second-order Taylor expansion, i.e., (7) **(c)** update from removing the second-order term from (7) **(d)** update from removing the first-order term from (7) **(e)** update from further adding softmax, i.e., (8)

B Different Manipulations to the Output Matrix

As mentioned in Section 3, CRATE replaces the output matrix $\boldsymbol{U} = [\boldsymbol{U}_1, \ldots, \boldsymbol{U}_K]$ in the MSSA operator with learnable $\boldsymbol{W}$ (which is different from $\boldsymbol{U}$). We then raise the following question on the manipulation of the output matrix: if we are free to adjust the output matrix while sacrificing interpretability, can we find more alternatives that can outperform CRATE-C or even CRATE? In practice, we have experimented with setting this matrix to an identity or fixed randomly initialized matrix, but only to discover that transpose performs best (Table 3). Therefore, CRATE-T is a feasible choice without introducing new parameters, which can be utilized to better understand the SRR principle and its connection to the performance.

We want to clarify that the analysis here intends to compare the variants with CRATE-C, not CRATE, because CRATE introduces learnable parameters that are less interpretable. We believe there are at least some interesting conclusions from the comparison: 1) CRATE-N achieves better performance by following the SRR principle more faithfully, shedding light on the connection of

Table 3: Top-1 accuracy for CRATE and its variants trained on CIFAR-10 from scratch (width $d = 384$).

Models	CRATE-C	CRATE-N	CRATE-T	CRATE	CRATE-Fix	CRATE-Identity
# Params	3.94M	3.94M	3.94M	5.71M	3.94M	3.94M
Accuracy	76.87	81.52	85.49	86.67	80.73	83.18

SRR to generalization; 2) We need to explore more design choices (e.g., CRATE-T, which may deviate from directly optimizing the SRR but still exhibit a similar architecture) to gain a complete understanding of the SRR principle for model performance (this motivates our Section 5).

C Experimental Details of Collecting Trained Models

Our experimental details to generate a family of trained models largely follow the previous work [18]. Models with heavy data augmentations tend to generalize better than those without them. It is therefore crucial to isolate the influence of data augmentations from the change of other hyperparameters. We choose to remove data augmentations during training to ensure that most models can be trained to meet the stopping criterion. We include Layer Normalization [2] before each operator during training, but also remove it when evaluating the complexity measures.

In this experiment, we vary across 5 sets of hyperparameters, i.e., batch size, initial learning rate, width, dropout probability, and model type. We present the choices of these hyperparameters in Table 4. Adam [21] is used as the default optimizer. Model depth is kept as $L = 12$ and number of subspaces $K = 6$. Dropout is applied after adding positional encoding, softmax function, and output projection in MSSA operator.

Table 4: Choices of hyperparameters.

Hyperparameters	Choices
batch size	$\{64, 128\}$
initial learning rate	$\{2 \times 10^{-5}, 1 \times 10^{-4}\}$
width	$\{384, 768\}$
dropout	$\{0.0, 0.1\}$
model type	$\{\text{CRATE-C}, \text{CRATE-N}, \text{CRATE-T}, \text{CRATE}\}$

D Correlation of Complexity Measures when width $d = 768$

Table 5 and Figure 6 give results on correlation to generalization when width $d = 768$. We see that SRR is slightly better than other baseline measures in terms of overall τ. In the axes of dropout and model type, however, it underperforms PAC-Bayes flatness measure. This implies that width could have a considerable influence on studying SRR as a complexity measure. We leave it for future work.

E Comparisons of Efficient Implementations

Table 6 compares different efficient implementations of SRR regularization. We find that randomly choosing layers to regularize generally worsens the performance. While regularizing shallower layers may bring more performance gain, leveraging the last layer already suffices to outperform the cross-entropy baseline. Specifying which layer to regularize could be expensive, especially when the model size grows. We opt for the last layer, which should be reasonable if depth scales. Our results indicate that this intuitive choice can already give consistent performance gains in different settings.

Table 5: Correlation of complexity measures with generalization gap (width $d = 768$).

	Batch size	Learning rate	Dropout	Model type	Overall τ	Ψ
ℓ_2-norm	0.000	-0.375	-0.625	-0.250	-0.310	-0.313
ℓ_2-norm-init	0.000	-0.375	-0.625	-0.208	-0.274	-0.302
# params	0.000	0.000	0.000	-0.295	-0.188	-0.074
1/margin	-0.125	0.375	0.625	-0.208	0.173	0.167
sum-of-spec	0.000	-0.375	-0.625	-0.375	-0.310	-0.344
prod-of-spec	0.000	-0.375	-0.625	-0.417	-0.339	-0.354
sum-of-spec/margin	0.000	-0.375	-0.625	-0.458	-0.319	-0.365
prod-of-spec/margin	0.000	-0.375	-0.625	-0.417	-0.327	-0.354
fro/spec	0.000	0.375	0.500	-0.083	0.242	0.239
spec-init-main	0.000	-0.375	-0.625	-0.417	-0.331	-0.354
spec-orig-main	0.000	-0.375	-0.625	-0.417	-0.331	-0.354
sum-of-fro	0.000	-0.375	-0.625	-0.333	-0.306	-0.333
prod-of-fro	0.000	-0.375	-0.625	-0.250	-0.278	-0.313
sum-of-fro/margin	-0.125	-0.375	-0.500	-0.167	-0.286	-0.292
prod-of-fro/margin	-0.125	-0.375	-0.500	-0.125	-0.238	-0.281
fro-distance	0.000	-0.375	-0.625	-0.208	-0.274	-0.302
spec-distance	0.000	-0.375	-0.625	-0.417	-0.322	-0.354
param-norm	0.000	-0.375	-0.625	-0.250	-0.310	-0.316
path-norm	-0.250	-0.625	0.125	-0.500	-0.415	-0.313
pac-bayes-init	0.000	-0.375	-0.625	0.250	-0.214	-0.188
pac-bayes-orig	0.000	0.375	0.625	0.167	0.315	0.292
$1/\sigma$ pac-bayes-flatness	0.000	0.375	0.688	0.573	0.337	**0.409**
SRR	0.125	0.500	0.250	0.375	**0.407**	0.313

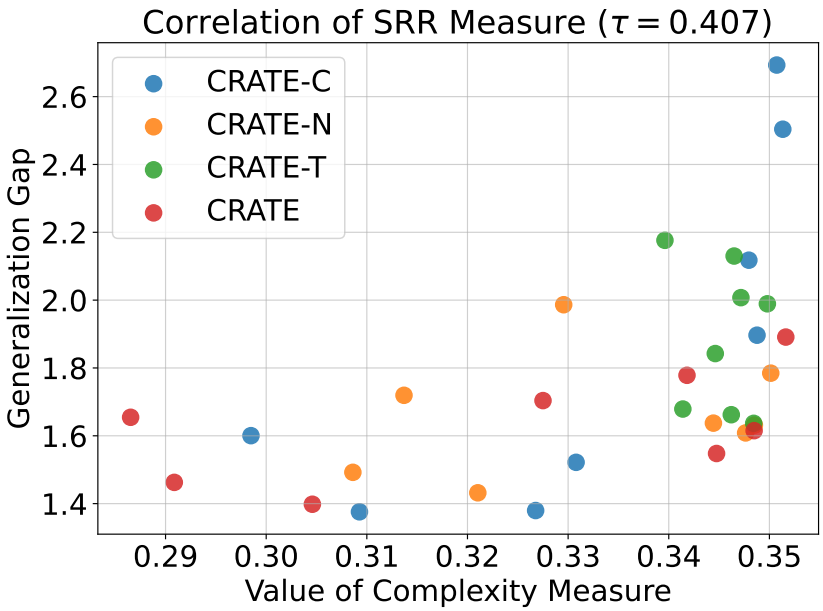


Figure 6: A scatter plot illustrating the value of SRR measure and generalization gap across CRATE and variants with network width $d = 768$.

Table 6: Top-1 accuracy for CRATE and its variants trained with efficient implementations of SRR regularization on CIFAR-10 from scratch (width $d = 384$).

Training methods	CIFAR-10			
	CRATE-C	CRATE-N	CRATE-T	CRATE
cross-entropy	76.87	81.52	85.49	86.67
+ Layer 2 reg	77.75	**82.41**	**85.84**	**87.03**
+ Layer 4 reg	**77.95**	81.57	85.46	87.03
+ Layer 6 reg	77.48	80.83	85.22	87.02
+ Layer 8 reg	77.04	81.29	85.12	86.64
+ Layer 10 reg	77.44	81.19	85.68	86.67
+ Layer 12 reg	77.61	81.91	85.52	86.79
+ Random layer reg	75.19	79.66	84.27	85.36

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We list our contributions in the introduction derived from the main sections.

   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]

   Justification: We provide an additional section discussing the limitations of our work in the manuscript.

   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: Our paper is mostly about empirical evaluation and therefore does not provide novel theories.

   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 faithfully describe our experimental details in the main text.

   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: Our paper builds heavily on previous works which are publicly available.

   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).
   - 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: All the detailed are stated in the main text.

   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: [No]

Justification: Due to lack of time, we do not include error bars in our experiment.

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 use Nvidia 3090 GPU for all the experiments.

   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 strictly follow the 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: [No]

    Justification: We believe this work makes preliminary efforts towards understanding a potentially principled way of designing Transformer-like models. But there are still gaps and limitations on the implementations of sparse rate reduction as a guide to model design. The conclusion of our paper is still not definitive enough to make strong societal impacts.

    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: This is no such risk.

    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: We have properly cited the most relevant papers which we build our code on.

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: There are no such assets introduced in the paper.

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: The paper does not involve crowdsourcing nor 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: The paper does not involve crowdsourcing nor 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.