SCHOOL OF COMPUTING & DATA SCIENCE
The University of Hong Kong

NEURAL INFORMATION PROCESSING SYSTEMS

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

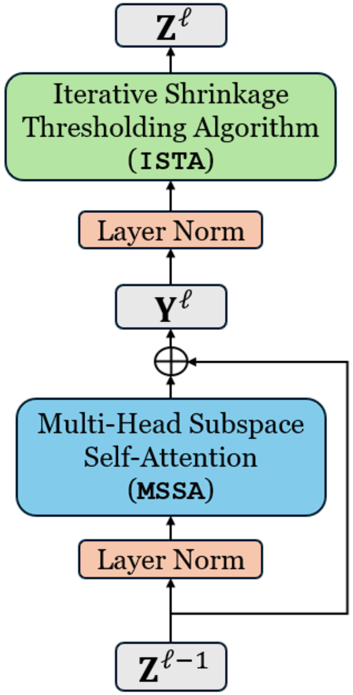
Yunzhe Hu, Difan Zou, and Dong Xu

The University of Hong Kong

Outline

- **Background**: Designing Transformer-like models via Sparse Rate Reduction (SRR)
- **Key Investigations**:
  - Analysis of behaviors of self-attention operator
  - Correlation between learning objective SRR and generalization
- **Main Results:**
  - SRR measure can be a strong predictor of generalization, better than sharpness
  - SRR measure can incorporated as regularization for improved performance
- **Takeaways**

Background



White-box Transformer **CRATE**
[Yu et al. NeurIPS 2023]

Sparse Rate Reduction (SRR):

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

$$R(\boldsymbol{Z}) \doteq \frac{1}{2}\log\det\left(\boldsymbol{I} + \frac{d}{N\epsilon^2}\boldsymbol{Z}^T\boldsymbol{Z}\right)$$

$$R^c(\boldsymbol{Z};\boldsymbol{U}) \doteq \sum_{k=1}^{K} R(\boldsymbol{U}_k^T\boldsymbol{Z})$$

Alternating Minimization:

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

$$\boldsymbol{Z}^{\ell} = \underbrace{\mathrm{ReLU}\left(\boldsymbol{Y}^{\ell} + \beta(\boldsymbol{D}^{\ell})^T(\boldsymbol{Y}^{\ell} - \boldsymbol{D}^{\ell}\boldsymbol{Y}^{\ell}) - \beta\lambda\mathbf{1}\right)}_{\text{minimize } \lambda\|\boldsymbol{Z}\|_0 - R(\boldsymbol{Z})}$$

$$\mathrm{MSSA}(\boldsymbol{Z};\boldsymbol{U}) = \sum_{k=1}^{K}\boldsymbol{U}_k\boldsymbol{U}_k^T\boldsymbol{Z}\,\mathrm{softmax}((\boldsymbol{U}_k^T\boldsymbol{Z})^T(\boldsymbol{U}_k^T\boldsymbol{Z}))$$

$$= \underbrace{[\boldsymbol{U}_1,\ldots,\boldsymbol{U}_K]}_{\boldsymbol{U}}\begin{bmatrix}\boldsymbol{U}_1^T\boldsymbol{Z}\,\mathrm{softmax}((\boldsymbol{U}_1^T\boldsymbol{Z})^T(\boldsymbol{U}_1^T\boldsymbol{Z}))\\ \vdots \\ \boldsymbol{U}_K^T\boldsymbol{Z}\,\mathrm{softmax}((\boldsymbol{U}_K^T\boldsymbol{Z})^T(\boldsymbol{U}_K^T\boldsymbol{Z}))\end{bmatrix}$$

Background

CRATE further introduces more parameters $\boldsymbol{W}$ at the expense of interpretability.

$$\operatorname{MSSA}(\boldsymbol{Z};\boldsymbol{U},\boldsymbol{W}) = \boldsymbol{W}\begin{bmatrix} \boldsymbol{U}_k^T\boldsymbol{Z}\,\operatorname{softmax}((\boldsymbol{U}_k^T\boldsymbol{Z})^T(\boldsymbol{U}_k^T\boldsymbol{Z})) \\ \vdots \\ \boldsymbol{U}_k^T\boldsymbol{Z}\,\operatorname{softmax}((\boldsymbol{U}_k^T\boldsymbol{Z})^T(\boldsymbol{U}_k^T\boldsymbol{Z})) \end{bmatrix}$$

To differentiate, we refer to the theoretically-driven framework as **CRATE-C(onceptual)**

$$\operatorname{MSSA}(\boldsymbol{Z};\boldsymbol{U}) = [\boldsymbol{U}_1,\ldots,\boldsymbol{U}_K]\begin{bmatrix} \boldsymbol{U}_k^T\boldsymbol{Z}\,\operatorname{softmax}((\boldsymbol{U}_k^T\boldsymbol{Z})^T(\boldsymbol{U}_k^T\boldsymbol{Z})) \\ \vdots \\ \boldsymbol{U}_k^T\boldsymbol{Z}\,\operatorname{softmax}((\boldsymbol{U}_k^T\boldsymbol{Z})^T(\boldsymbol{U}_k^T\boldsymbol{Z})) \end{bmatrix}$$

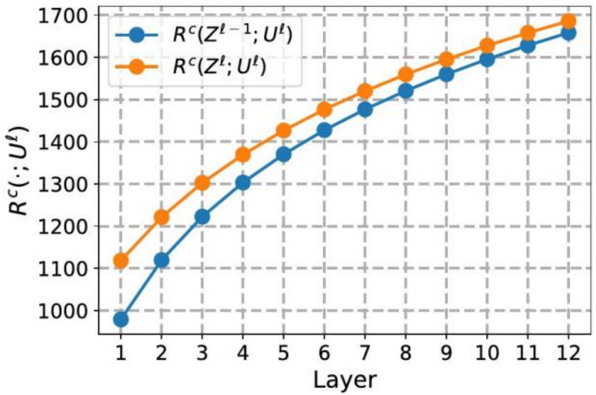
Pitfalls of Building CRATE-C

Given that MSSA operator is designed to minimize $R^c(Z;U)$, it is supposed to **decrease** monotonically as layer goes deeper.

$$\boldsymbol{Z}^{\ell} = \boldsymbol{Z}^{\ell-1} + \sum_{k=1}^{K} \boldsymbol{U}_k^{\ell} {\boldsymbol{U}_k^{\ell}}^T \boldsymbol{Z}^{\ell-1} \operatorname{softmax}(({\boldsymbol{U}_k^{\ell}}^T \boldsymbol{Z}^{\ell-1})^T ({\boldsymbol{U}_k^{\ell}}^T \boldsymbol{Z}^{\ell-1}))$$

<u>Does the operation really achieve its design goal?</u>

- **Empirically, No.** We can show in an isolated toy experiment that the update actually yields a counterproductive effect.



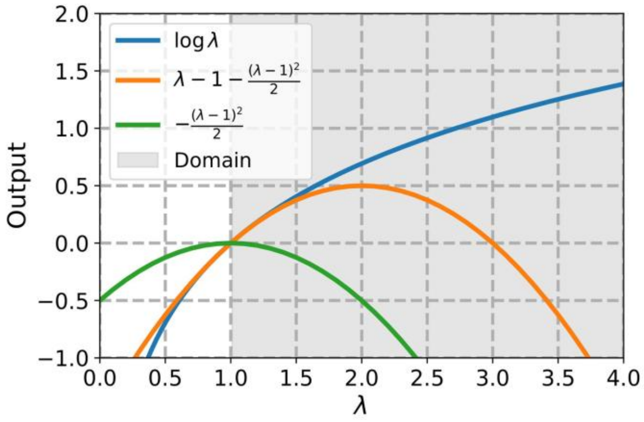
- **Theoretically, No.** We can also reveal this derivation artifacts from the eigenvalue perspective (see slides later).

Revisit and Interpret the Derivation

We first rewrite R^c with eigenvalues λ_i^k $(i = 1, \dots N)$ of $\boldsymbol{I} + \gamma\left(\boldsymbol{U}_k^T\boldsymbol{Z}\right)^T \boldsymbol{U}_k^T\boldsymbol{Z}$. Not that $\lambda_i^k \geq 1$. Then we show R^c can be lower bounded by its Taylor expansions.

$$R^c(\boldsymbol{Z};\boldsymbol{U}) = \sum_{k=1}^{K}\sum_{i=1}^{N}\frac{1}{2}\log\lambda_i^k \geq \sum_{k=1}^{K}\sum_{i=1}^{N}\frac{1}{2}\left(\lambda_i^k - 1 - \frac{(\lambda_i^k-1)^2}{2}\right)$$

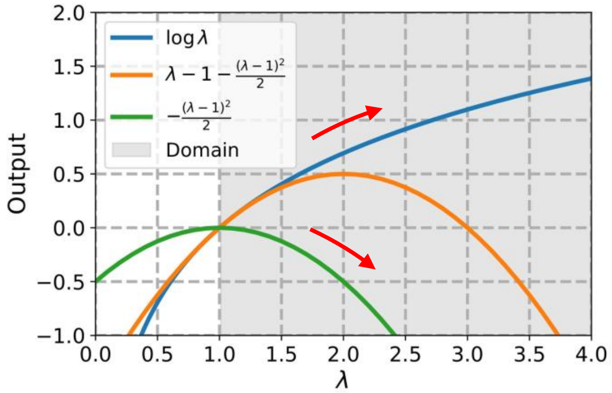
$$= \sum_{k=1}^{K}\left(\underbrace{\frac{\gamma}{2}\|\boldsymbol{U}_k^T\boldsymbol{Z}\|_F^2}_{\text{First-order term}} \underbrace{-\frac{\gamma^2}{4}\|(\boldsymbol{U}_k^T\boldsymbol{Z})^T\boldsymbol{U}_k^T\boldsymbol{Z}\|_F^2}_{\text{Second-order term}}\right)$$



Revisit and Interpret the Derivation

MSSA operator with skip connection is constructed by performing an **approximation** of gradient descent on R^c.

$$
\begin{aligned}
\boldsymbol{Z} - \alpha \nabla_{\boldsymbol{Z}} R^c(\boldsymbol{Z}; \boldsymbol{U}) &= \boldsymbol{Z} - \alpha\gamma \sum_{k=1}^{K} \boldsymbol{U}_k \boldsymbol{U}_k^T \boldsymbol{Z} \left(\boldsymbol{I} + \gamma (\boldsymbol{U}_k^T \boldsymbol{Z})^T (\boldsymbol{U}_k^T \boldsymbol{Z}) \right)^{-1} \\
&\approx \boldsymbol{Z} - \alpha \left(\underbrace{\gamma \sum_{k=1}^{K} \boldsymbol{U}_k \boldsymbol{U}_k^T \boldsymbol{Z}}_{\nabla \text{ of first-order term}} \underbrace{- \gamma^2 \sum_{k=1}^{K} \boldsymbol{U}_k \boldsymbol{U}_k^T \boldsymbol{Z} (\boldsymbol{U}_k^T \boldsymbol{Z})^T (\boldsymbol{U}_k^T \boldsymbol{Z})}_{\nabla \text{ of second-order term}} \right) \\
&\approx \boldsymbol{Z} + \alpha\gamma^2 \sum_{k=1}^{K} \boldsymbol{U}_k \boldsymbol{U}_k^T \boldsymbol{Z} \, \mathrm{softmax}((\boldsymbol{U}_k^T \boldsymbol{Z})^T (\boldsymbol{U}_k^T \boldsymbol{Z})).
\end{aligned}
$$



Why this construction produces the opposite effect, i.e., increasing R^c ? Only utilizing the <u>second-order term</u> of its gradient !

Variants of CRATE

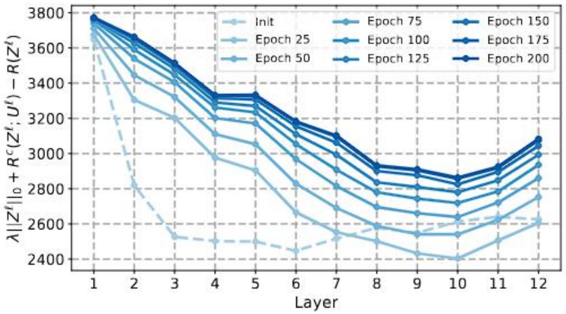
To implement the design purpose more faithfully, the sign before MSSA operator can be naturally reversed, performing ascent method. We name this framework **CRATE-N(egative).**

$$\boldsymbol{Z} - \alpha\gamma^2 \sum_{k=1}^{K} \boldsymbol{U}_k \boldsymbol{U}_k^T \boldsymbol{Z} \,\mathrm{softmax}((\boldsymbol{U}_k^T \boldsymbol{Z})^T (\boldsymbol{U}_k^T \boldsymbol{Z}))$$

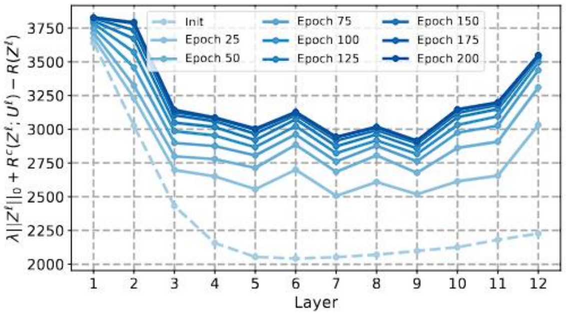
Can we further find a variant that performs competitively with CRATE without new parameters ? Well, a simple transpose could do (see experiments later). We term this one **CRATE-T(ranspose).**

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

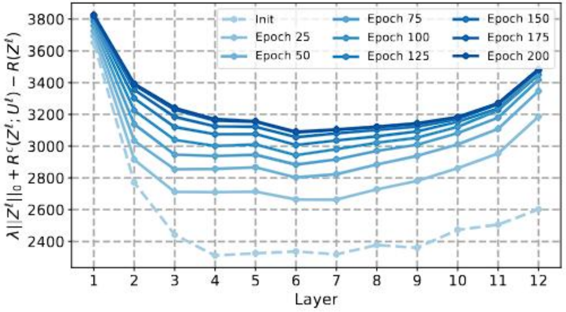
Behaviors of Sparse Rate Reduction



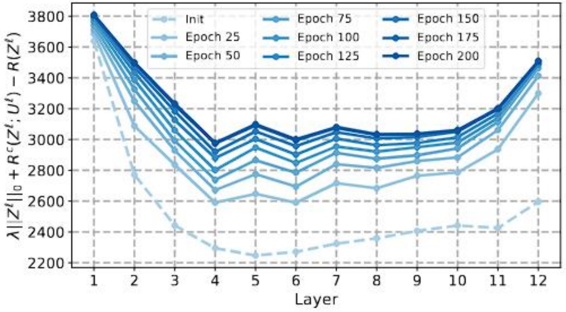
(a) Sparse rate reduction measure of CRATE-C (7)



(b) Sparse rate reduction measure of CRATE-N (8)



(c) Sparse rate reduction measure of CRATE-T (9)



(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.

Whether Sparse Rate Reduction Benefits Generalization?

So far, we have partially confirmed the validity of different implementations of transformer-like models.

However, there are still some lingering questions

- Whether this SRR objective is beneficial or principled for these models to generalize ?
- If so, how much is the benefit ?

Whether Sparse Rate Reduction Benefits Generalization?

We will explore its causal relationship to the generalization and adopt SRR as an empirical predictor of generalization (measure of complexity).

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

Typically, a good measure should have the property where lower complexity should indicate smaller generalization gap. For example, the following is true if the measure μ is described by generalization bound.

$$L_{\text{test}} - \hat{L}_{\text{train}} \leq \sqrt{\frac{\mu}{m}}$$

Correlation Analysis

Similar to [Jiang et al. ICLR 2020], we collected a set of models with varied hyper-parameters trained until convergence and evaluated how well the generalization gap correlates with the measure.

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

Table 3: 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	{CRATE, CRATE-C, CRATE-N, CRATE-T}

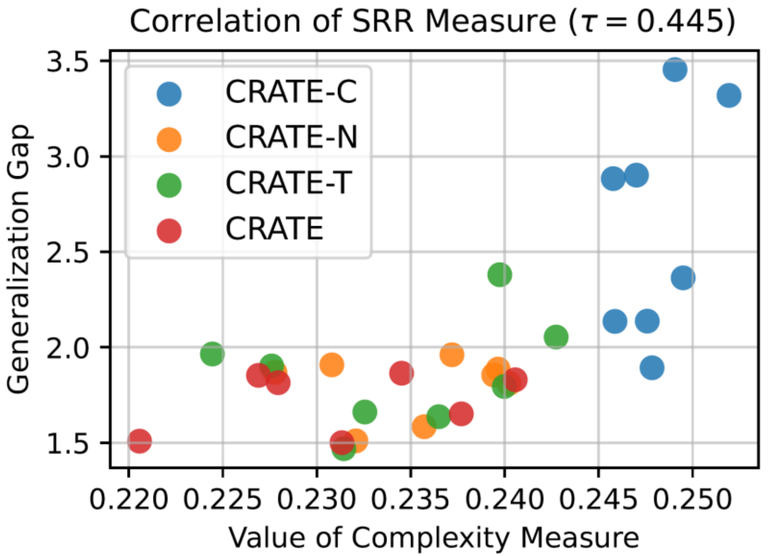
Kendall's rank-correlation coefficient: Range in $[-1,1]$. The close to one, the stronger the positive correlation.

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

Correlation Analysis Results

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**



Better predictive power than widely investigated flatness-based measure

Sparse Rate Reduction as Regularization

Since SRR measure enjoys a strong correlation to generalization, it is reasonable to incorporate it during training and optimize it with task-specific loss simultaneously, similar to sharpness-aware minimization [Foret et al. ICLR 2021] for improved generalization.

Well, the most straightforward way is through regularization.

$$\min_{\boldsymbol{w}} \mathcal{L}_{\text{ce}}(\boldsymbol{w}) + \eta \cdot \frac{1}{L} \sum_{\ell=1}^{L} \mu_{\text{SRR}}^{\ell}(\boldsymbol{w}^{\ell}; \boldsymbol{Z}_{\text{StopGrad}}^{\ell})$$

Sparse Rate Reduction as Regularization Results

An efficient implementation at the last layer can already give consistent performance gain.

In fact, we also find that imposing regularization on first few layers performs best. Only as a proof-of-concept for scalable depth here.

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**

Table 5: 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

Takeaways

- SRR objective can be viewed as an energy function that is optimized in the forward pass of transformer-like models.
    - It *almost* monotonically decreases and hovers around the stationary point.
    - Behaviors persist across varied implementations.

- SRR objective could be a choice to design transformer-like models, but not necessarily principled.
    - We demonstrate its *positive and strong (by comparison)* correlation with generalization.
    - More faithful instantiation does not necessarily give a better model.