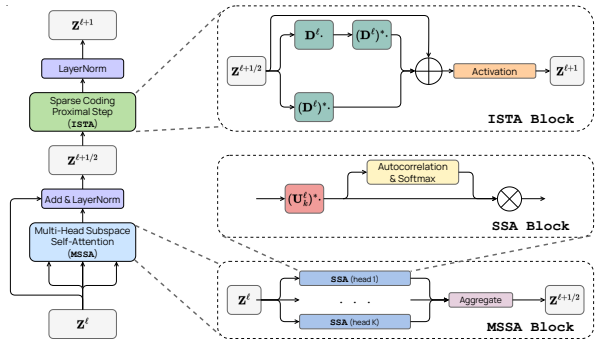
White-Box Transformers via Sparse Rate Reduction

Yaodong Yu, Sam Buchanan, Druv Pai,
Tianzhe Chu, Ziyang Wu, Shengbang Tong,
Benjamin D. Haeffele, Yi Ma

June 10, 2023

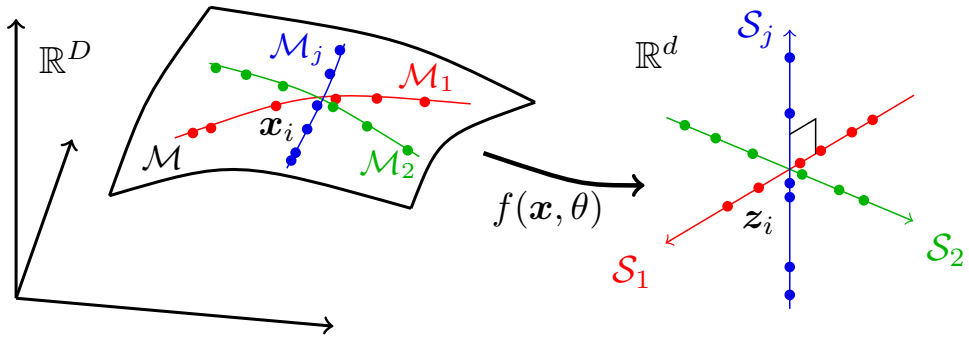


Outline

Identification/Representation of High-Dim Structured Data

Focus on one half of our goal:

Given samples
$\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m] \subset \cup_{j=1}^{k} \mathcal{M}_j$,
seek a good representation
$\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m] \subset \mathbb{R}^d$
through a continuous mapping:
$f(\boldsymbol{x}, \boldsymbol{\theta}) : \boldsymbol{x} \in \mathbb{R}^D \mapsto \boldsymbol{z} \in \mathbb{R}^d$.

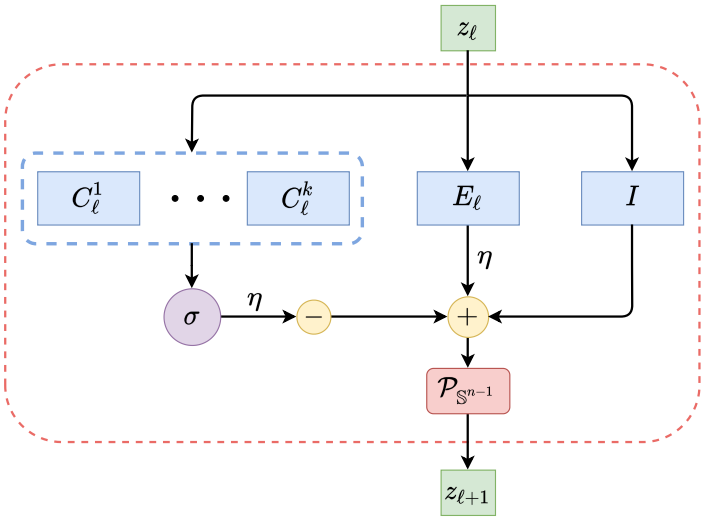


How to obtain a white-box architecture f that simultaneously identifies and represents large-scale datasets?

Recap: White-Box Deep Networks

A promising approach: signal models $\Longrightarrow$ deep architectures

- Convolutional sparse coding networks [Papyan et al. 2018]
- Scattering networks [Bruna & Mallat 2013]
- ReduNets [Chan, Yu et al. 2022]



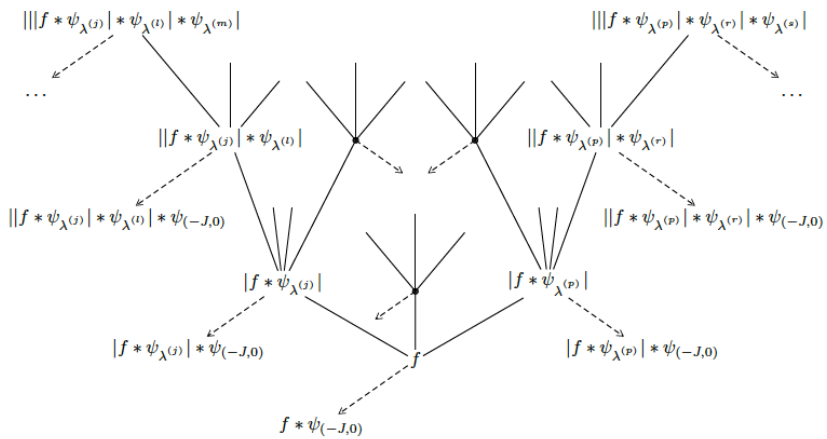


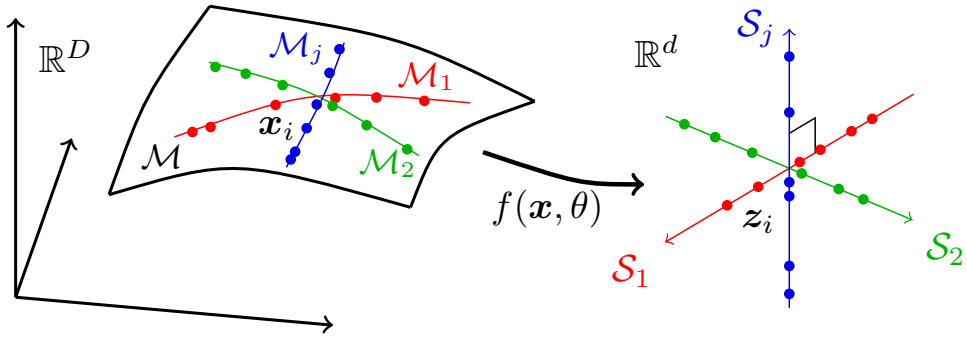
Fig. 2: Scattering network architecture based on wavelet filters and the modulus non-linearity. The elements of the feature vector $\Phi_W(f)$ in (1) are indicated at the tips of the arrows.

Figure: Left: **ReduNet** layer. Right: **Scattering Network** [Bruna & Mallat 2013] [Wiatowski & Bölcskei 2018] (**only 2-3 layers**).

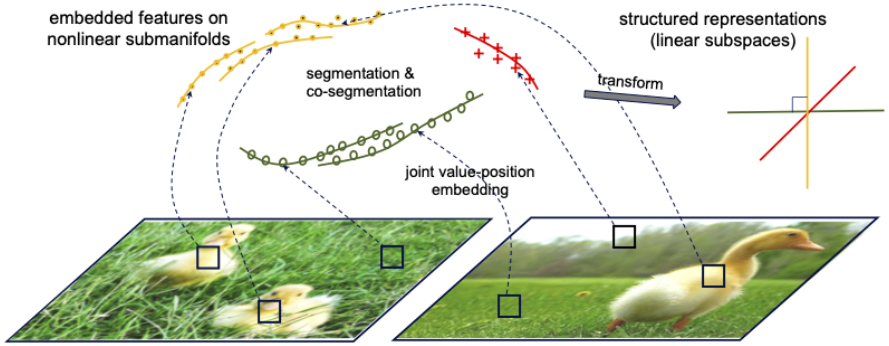
Pitfall of existing methods: Challenging to scale to massive datasets with strong performance

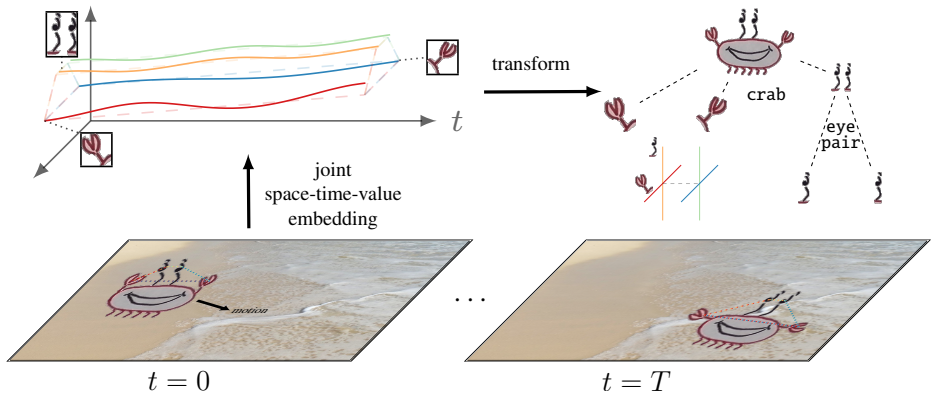
Improved White-Box Scaling by Improved Signal Modeling?

So far: *Each* **sample** *is drawn from a mixture of manifolds*



Better? *Each sample* $\supset$ **correlated tokens**—*mixture of manifold marginals!*

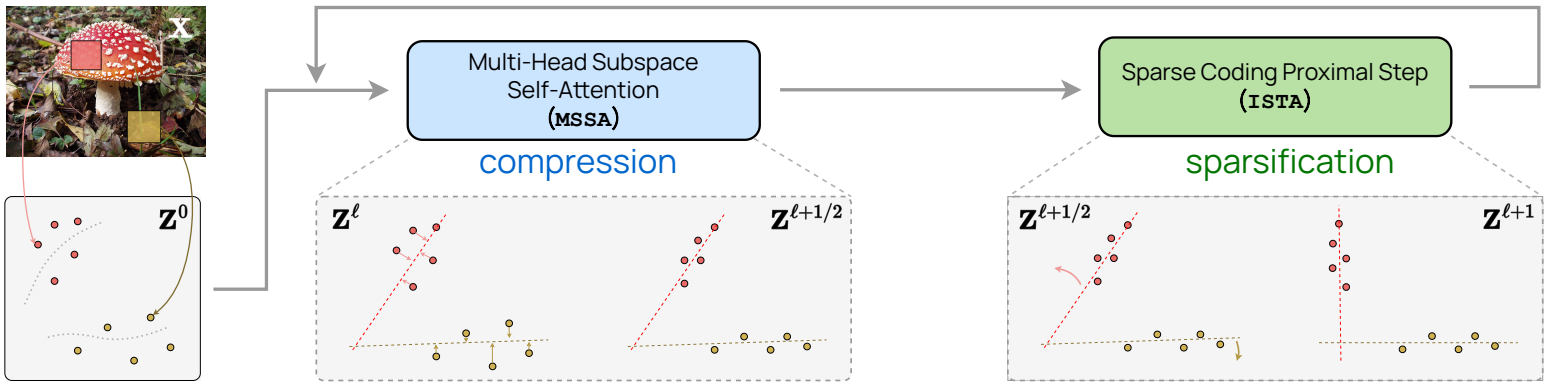




CRATE: A White-Box Transformer via Sparse MCR^2

A white-box, mathematically interpretable, transformer-like deep network architecture from **iterative unrolling** optimization schemes to incrementally optimize the sparse rate reduction objective:

$$\max_{f \in \mathcal{F}} \mathbb{E}_{\boldsymbol{Z}} \left[\Delta R(\boldsymbol{Z}; \boldsymbol{U}_{[K]}) - \|\boldsymbol{Z}\|_0 \right], \quad \boldsymbol{Z} = f(\boldsymbol{X}).$$



CRATE: White-Box Transformers via Sparse Rate Reduction

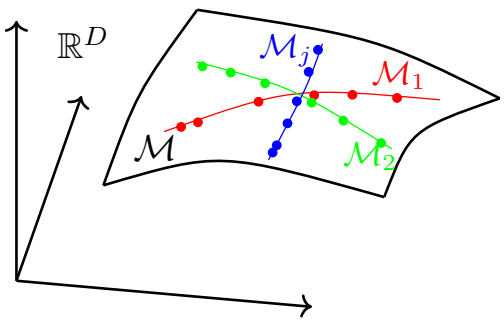
https://arxiv.org/abs/2306.01129

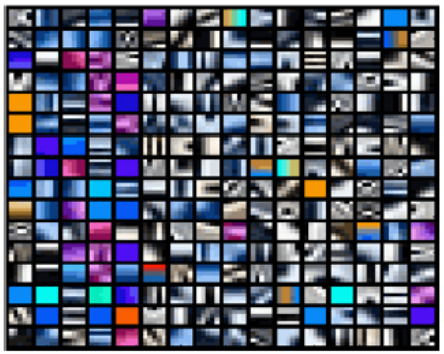
Sparse MCR^2 Objective and Incremental Representation

The sparse rate reduction (Sparse MCR^2) objective is defined as

$$\begin{aligned}&\arg\max_{f\in\mathcal{F}} \mathbb{E}_{\boldsymbol{Z}}\left[\Delta R(\boldsymbol{Z};\boldsymbol{U}_{[K]}) - \|\boldsymbol{Z}\|_0\right]\\ =&\arg\min_{f\in\mathcal{F}} \mathbb{E}_{\boldsymbol{Z}}\Big[\underbrace{R^c(\boldsymbol{Z};\boldsymbol{U}_{[K]})}_{\text{compression}} + \underbrace{\|\boldsymbol{Z}\|_0 - R(\boldsymbol{Z})}_{\text{sparsification}}\Big].\end{aligned}$$

$\boldsymbol{U}_{[K]} = (\boldsymbol{U}_1, \ldots, \boldsymbol{U}_K)$, $\boldsymbol{U}_k \in \mathbb{R}^{d\times p}$ are *subspaces parameterizing the marginal distribution of tokens* $(\boldsymbol{z}_i)_{i=1}^N$





Sparse MCR^2 Objective and Incremental Representation

The sparse rate reduction (Sparse MCR^2) objective is defined as

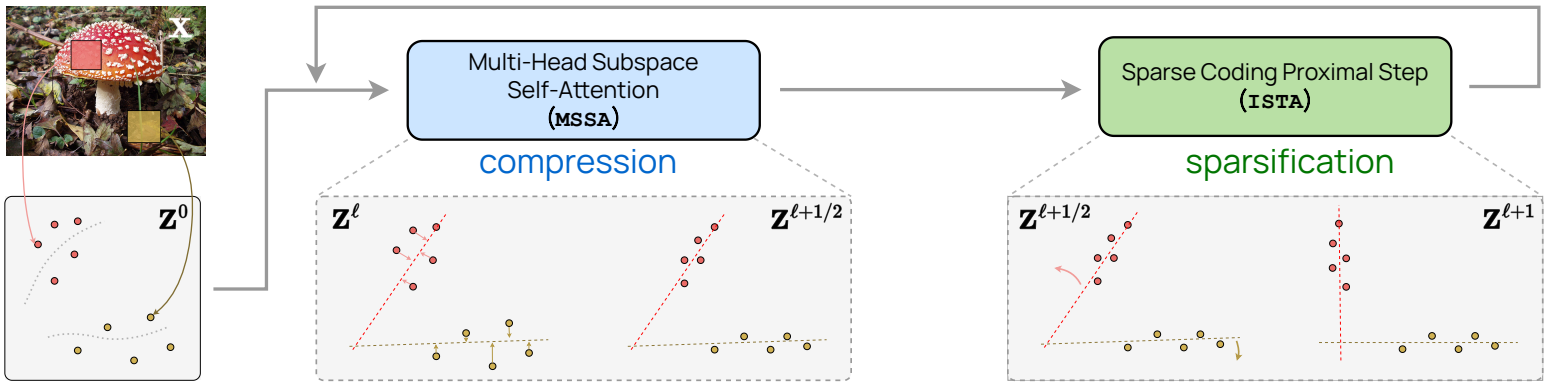
$$\max_{f \in \mathcal{F}} \mathbb{E}_{\boldsymbol{Z}} \left[\Delta R(\boldsymbol{Z}; \boldsymbol{U}_{[K]}) - \|\boldsymbol{Z}\|_0 \right]$$

$$= \min_{f \in \mathcal{F}} \mathbb{E}_{\boldsymbol{Z}} \Big[\underbrace{R^c(\boldsymbol{Z}; \boldsymbol{U}_{[K]})}_{\text{compression}} + \underbrace{\|\boldsymbol{Z}\|_0 - R(\boldsymbol{Z})}_{\text{sparsification}} \Big].$$

The global transformation f is realized through **local transformations**:

$$f : \boldsymbol{X} \xrightarrow{f^0} \boldsymbol{Z}^0 \to \cdots \to \boldsymbol{Z}^\ell \xrightarrow{f^\ell} \boldsymbol{Z}^{\ell+1} \to \cdots \to \boldsymbol{Z}^L = \boldsymbol{Z}.$$

Each f^ℓ deforms $\boldsymbol{Z}^\ell$ according to its own **local signal model** $\boldsymbol{U}^\ell_{[K]}$.

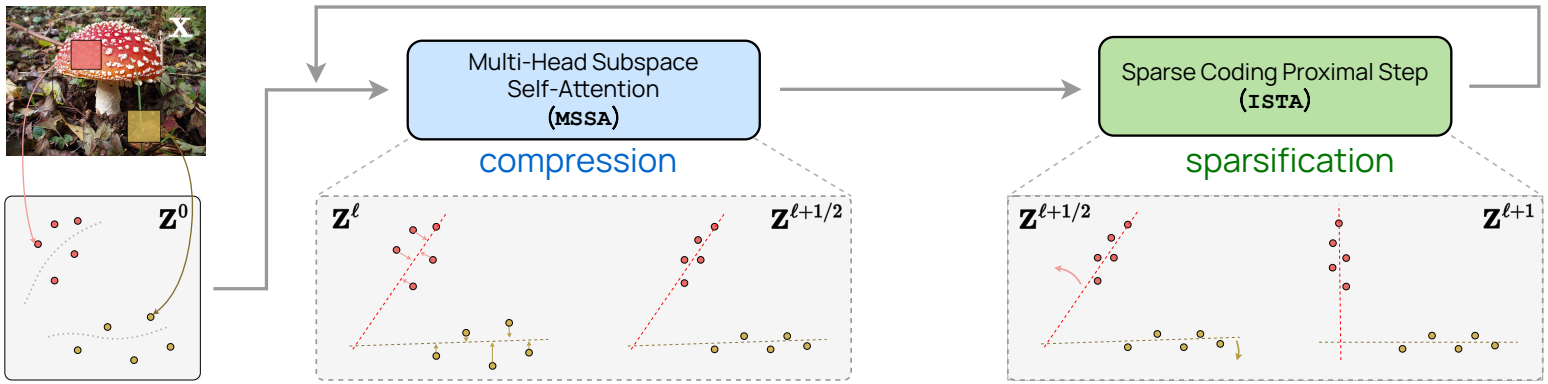


Sparse MCR^2 Objective and Incremental Representation

The sparse rate reduction (Sparse MCR^2) objective is defined as

$$\max_{f \in \mathcal{F}} \mathbb{E}_{\boldsymbol{Z}} \left[\Delta R(\boldsymbol{Z}; \boldsymbol{U}_{[K]}) - \|\boldsymbol{Z}\|_0 \right]$$

$$= \min_{f \in \mathcal{F}} \mathbb{E}_{\boldsymbol{Z}} \Big[\underbrace{R^c(\boldsymbol{Z}; \boldsymbol{U}_{[K]})}_{\text{compression}} + \underbrace{\|\boldsymbol{Z}\|_0 - R(\boldsymbol{Z})}_{\text{sparsification}} \Big].$$



How to construct a representation f to incrementally optimize the compression term and the sparsification term?

Recap: Compression and Expansion in MCR^2

Expansion:

$$R(\boldsymbol{Z}) = \frac{1}{2}\sum_{k=1}^{K} \text{logdet}\left(\boldsymbol{I} + \frac{d}{N\epsilon^2}\boldsymbol{Z}^*\boldsymbol{Z}\right)$$

Compression:

$$R^c(\boldsymbol{Z};\boldsymbol{U}_{[K]}) = \frac{1}{2}\sum_{k=1}^{K} \text{logdet}\left(\boldsymbol{I} + \frac{p}{N\epsilon^2}(\boldsymbol{U}_k^*\boldsymbol{Z})^*(\boldsymbol{U}_k^*\boldsymbol{Z})\right)$$

Gradient of Rate Distortion:

$$\left.\frac{\partial R(\boldsymbol{Z})}{\partial \boldsymbol{Z}}\right|_{\boldsymbol{Z}_\ell} = \underbrace{\alpha(\boldsymbol{I}+\alpha\boldsymbol{Z}_\ell\boldsymbol{Z}_\ell^*)^{-1}\boldsymbol{Z}_\ell}_{\text{auto-regress residual}} \approx \underbrace{\alpha\big[\boldsymbol{Z}_\ell - \alpha\boldsymbol{Z}_\ell(\boldsymbol{Z}_\ell^*\boldsymbol{Z}_\ell)\big]}_{\text{self-attention head}}.$$

Compression in Sparse MCR^2

To optimize the compression term $R^c(\boldsymbol{Z}; \boldsymbol{U}_{[K]})$, we propose to compress the set of tokens against the subspaces $(\boldsymbol{U}_k)_{k=1}^K$ by minimizing the coding rate via "approximate" gradient descent

$$\begin{aligned}\text{(Gradient Descent):}\quad & \boldsymbol{Z}^\ell - \kappa \nabla_{\boldsymbol{Z}} R^c(\boldsymbol{Z}^\ell; \boldsymbol{U}_{[K]}) \\ & \approx \left(1 - \kappa \cdot \frac{p}{N\epsilon^2}\right) \boldsymbol{Z}^\ell + \kappa \cdot \frac{p}{N\epsilon^2} \cdot \texttt{MSSA}(\boldsymbol{Z}^\ell | \boldsymbol{U}_{[K]}),\end{aligned}$$

where MSSA is defined through an SSA operator as:

$$\texttt{SSA}(\boldsymbol{Z}|\boldsymbol{U}_k) = (\boldsymbol{U}_k^* \boldsymbol{Z})\,\mathsf{softmax}((\boldsymbol{U}_k^* \boldsymbol{Z})^* (\boldsymbol{U}_k^* \boldsymbol{Z})),$$

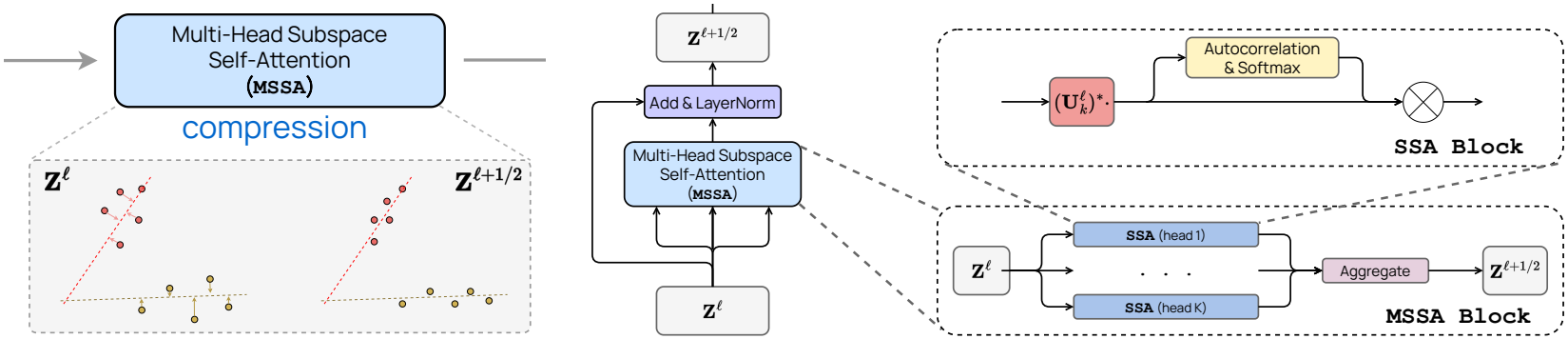
$$\texttt{MSSA}(\boldsymbol{Z}|\boldsymbol{U}_{[K]}) = \frac{p}{N\epsilon^2} \cdot [\boldsymbol{U}_1, \ldots, \boldsymbol{U}_K] \begin{bmatrix} \texttt{SSA}(\boldsymbol{Z}|\boldsymbol{U}_1) \\ \vdots \\ \texttt{SSA}(\boldsymbol{Z}|\boldsymbol{U}_K) \end{bmatrix}.$$

No need for separate query-Q, key-K, value-V in transformer attention block.

Compression in Sparse MCR^2

To optimize the compression term $R^c(\boldsymbol{Z}; \boldsymbol{U}_{[K]})$, we propose to compress the set of tokens against the subspaces $(\boldsymbol{U}_k)_{k=1}^K$ by minimizing the coding rate via "approximate" gradient descent

$$\boldsymbol{Z}^{\ell+1/2} = \boldsymbol{Z}^{\ell} + \texttt{MSSA}(\boldsymbol{Z}^{\ell} | \boldsymbol{U}_{[K]}).$$



(a) (b)

Figure: (a). Visualization of MSSA block; (b). Architecture of MSSA block.

Sparsification in Sparse MCR^2

To optimize the sparsification term $\|\boldsymbol{Z}\|_0 - R(\boldsymbol{Z})$, we posit a incoherent or orthogonal dictionary $\boldsymbol{D} \in \mathbb{R}^{d\times d}$ and sparsify $\boldsymbol{Z}^{\ell+1/2}$ with respect to $\boldsymbol{D}$, that is

$$\boldsymbol{Z}^{\ell+1/2} = \boldsymbol{D}\boldsymbol{Z}^{\ell+1}.$$

By the incoherence assumption, we have $\boldsymbol{D}^*\boldsymbol{D} \approx \boldsymbol{I}_d$; thus

$$R(\boldsymbol{Z}^{\ell+1}) \approx R(\boldsymbol{D}\boldsymbol{Z}^{\ell+1}) = R(\boldsymbol{Z}^{\ell+1/2}).$$

Thus we approximately optimize the sparsification objective with the following program:

$$\boldsymbol{Z}^{\ell+1} = \text{argmin}_{\boldsymbol{Z}}\|\boldsymbol{Z}\|_0 \quad \text{subject to} \quad \boldsymbol{Z}^{\ell+1/2} = \boldsymbol{D}\boldsymbol{Z}.$$

Sparsification in Sparse MCR^2

Given the sparse representation program

$$\boldsymbol{Z}^{\ell+1} = \text{argmin}_{\boldsymbol{Z}} \|\boldsymbol{Z}\|_0 \quad \text{subject to} \quad \boldsymbol{Z}^{\ell+1/2} = \boldsymbol{D}\boldsymbol{Z}.$$

we can relax it to an convex program, i.e., positive sparse coding:

$$\boldsymbol{Z}^{\ell+1} = \underset{\boldsymbol{Z}\geq 0}{\arg\min} \left[\lambda\|\boldsymbol{Z}\|_1 + \|\boldsymbol{Z}^{\ell+1/2} - \boldsymbol{D}\boldsymbol{Z}\|_F^2\right].$$

We can incrementally optimize the above objective by performing an unrolled proximal gradient descent step, known as an ISTA step:

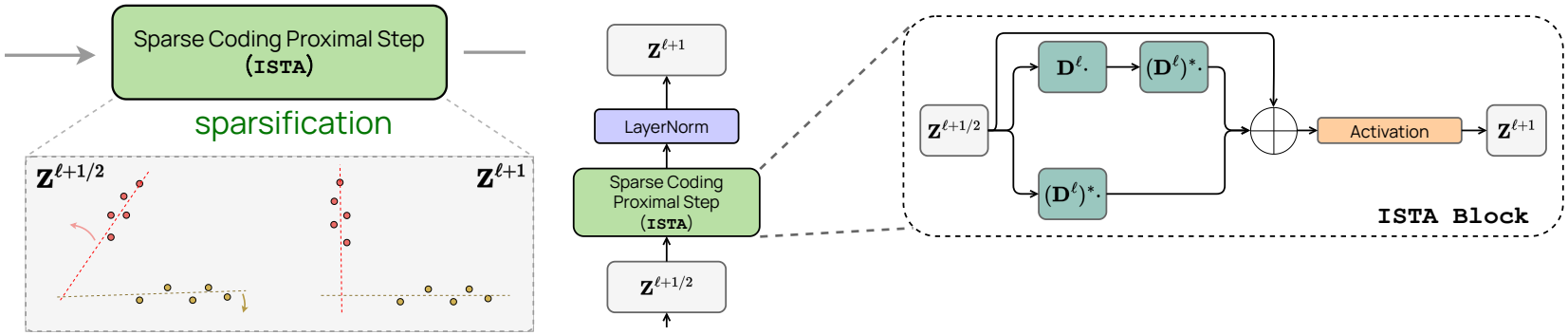
$$\begin{aligned}\boldsymbol{Z}^{\ell+1} &= \text{ReLU}(\boldsymbol{Z}^{\ell+1/2} + \eta\boldsymbol{D}^*(\boldsymbol{Z}^{\ell+1/2} - \boldsymbol{D}\boldsymbol{Z}^{\ell+1/2}) - \eta\lambda\mathbf{1}) \\ &:= \texttt{ISTA}(\boldsymbol{Z}^{\ell+1/2} \mid \boldsymbol{D}^\ell).\end{aligned}$$

The ISTA block uses much fewer parameters than transformer MLP block, and provides more interpretable representations.

Sparsification in Sparse MCR^2

To optimize the sparsification term $\|\boldsymbol{Z}\|_0 - R(\boldsymbol{Z})$, we propose to apply an unrolled proximal gradient descent step, known as an ISTA step:

$$\begin{aligned}\boldsymbol{Z}^{\ell+1} &= \mathrm{ReLU}(\boldsymbol{Z}^{\ell+1/2} + \eta \boldsymbol{D}^*(\boldsymbol{Z}^{\ell+1/2} - \boldsymbol{D}\boldsymbol{Z}^{\ell+1/2}) - \eta\lambda\mathbf{1}) \\ &:= \texttt{ISTA}(\boldsymbol{Z}^{\ell+1/2} \mid \boldsymbol{D}^\ell).\end{aligned}$$



(a) (b)

Figure: (a). Visualization of ISTA block; (b). Architecture of ISTA block.

One Layer of CRATE

Each layer of **CRATE** thus incrementally optimizes the compression term $R^c(\boldsymbol{Z}; \boldsymbol{U}_{[K]})$ and sparsification term $\|\boldsymbol{Z}\|_0 - R(\boldsymbol{Z})$,

$$\boldsymbol{Z}^{\ell+1} = f^\ell(\boldsymbol{Z}^\ell) = \texttt{ISTA}\Big(\underbrace{(\texttt{Id} + \texttt{MSSA})(\boldsymbol{Z}^\ell)}_{\boldsymbol{Z}^{\ell+1/2}}\Big).$$

More specifically,

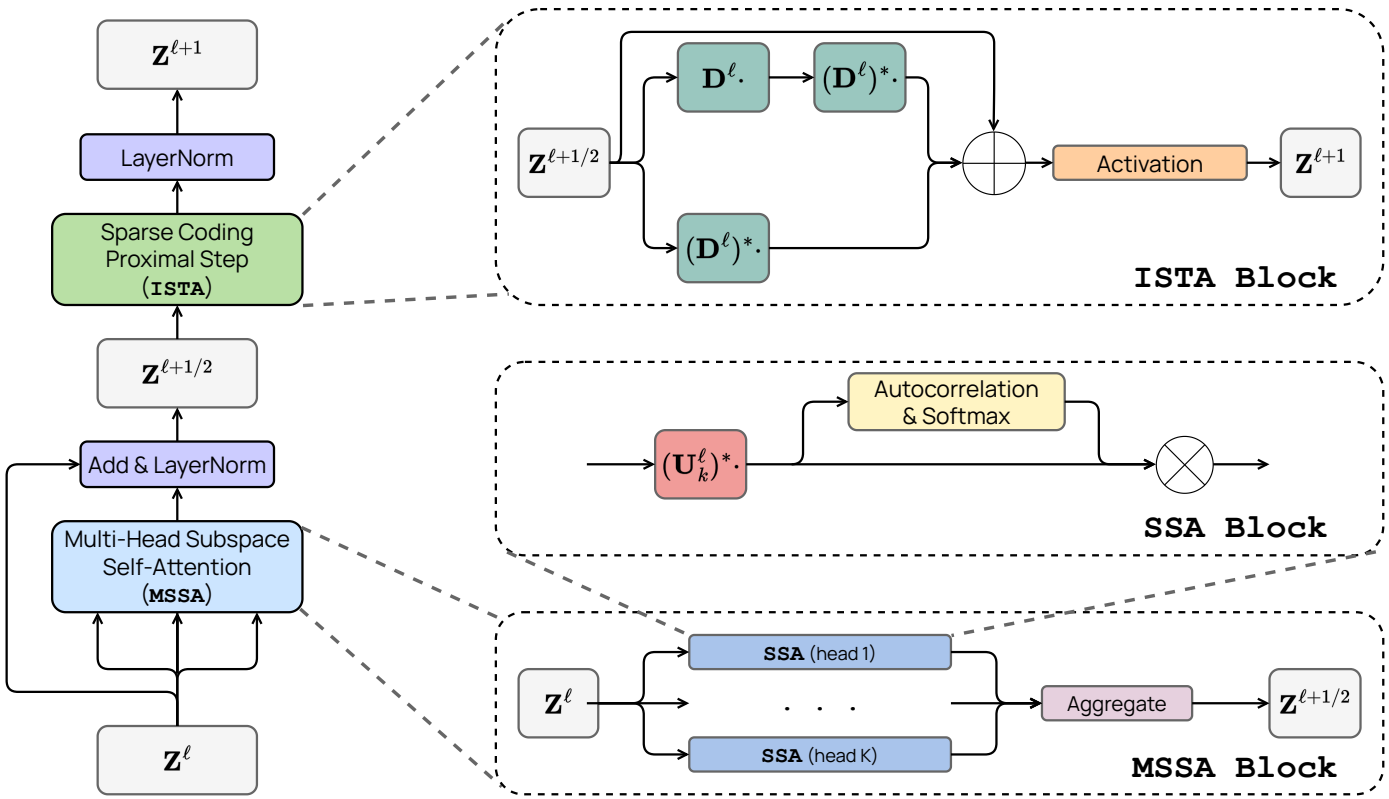
$$\boldsymbol{Z}^{\ell+1/2} = \boldsymbol{Z}^\ell + \texttt{MSSA}(\boldsymbol{Z}^\ell \,|\, \boldsymbol{U}^\ell_{[K]}), \qquad \text{[Compression step]}$$

$$\boldsymbol{Z}^{\ell+1} = \texttt{ISTA}(\boldsymbol{Z}^{\ell+1/2} \,|\, \boldsymbol{D}^\ell), \qquad \text{[Sparsification step]}$$

so the ℓ-th layer of the global representation f is

$$f^\ell : \boldsymbol{Z}^\ell \xrightarrow{\texttt{Id+MSSA}} \boldsymbol{Z}^{\ell+1/2} \xrightarrow{\texttt{ISTA}} \boldsymbol{Z}^{\ell+1}.$$

Overall White-Box CRATE Architecture



- Forward optimization: perform compression and sparsification.
- Learning from data: apply SGD to learn $(\boldsymbol{U}_{[K]}^{\ell}, \boldsymbol{D}^{\ell})_{\ell=1}^{L}$ from data.

Experiment I: Supervised Learning on ImageNet-1K

Experimental setup: let the CLS token of $\boldsymbol{Z}^L$ (i.e., the output token set of the last layer), and then apply a linear linear to perform supervised learning on ImageNet-1K using our proposed CRATE architecture.

Table 1: Top 1 accuracy of CRATE on various datasets with different model scales when pre-trained on ImageNet. For ImageNet/ImageNetReaL, we directly evaluate the top-1 accuracy. For other datasets, we use models that are pre-trained on ImageNet as initialization and the evaluate the transfer learning performance via fine-tuning.

Datasets	CRATE-T	CRATE-S	CRATE-B	CRATE-L	ViT-T	ViT-S
# parameters	6.09M	13.12M	22.80M	77.64M	5.72M	22.05M
ImageNet	66.7	69.2	70.8	71.3	71.5	72.4
ImageNet ReaL	74.0	76.0	76.5	77.4	78.3	78.4

- CRATE demonstrates promising performance on the ImageNet-1K dataset, indicating its potential for further advancement.

Experiment I: Supervised Learning on ImageNet-1K

Experimental setup: apply the CRATE model pre-trained on ImageNet-1K as initialization, and then evaluate transfer learning performance via fine-tuning.

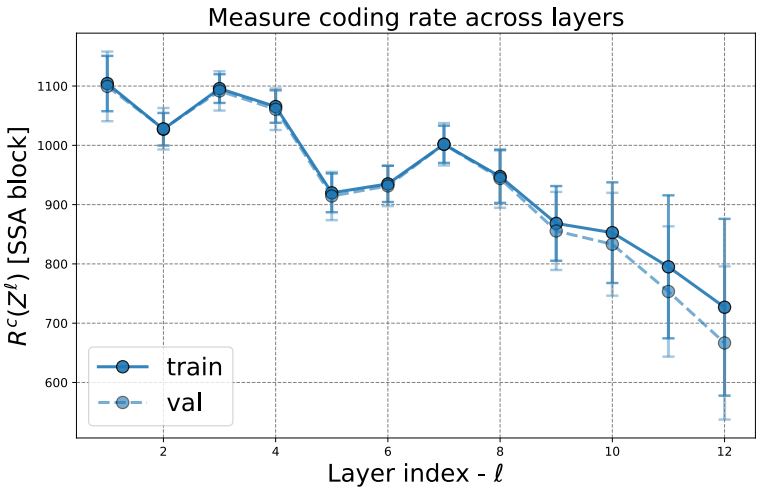
Table 1: Top 1 accuracy of CRATE on various datasets with different model scales when pre-trained on ImageNet. For ImageNet/ImageNetReaL, we directly evaluate the top-1 accuracy. For other datasets, we use models that are pre-trained on ImageNet as initialization and the evaluate the transfer learning performance via fine-tuning.

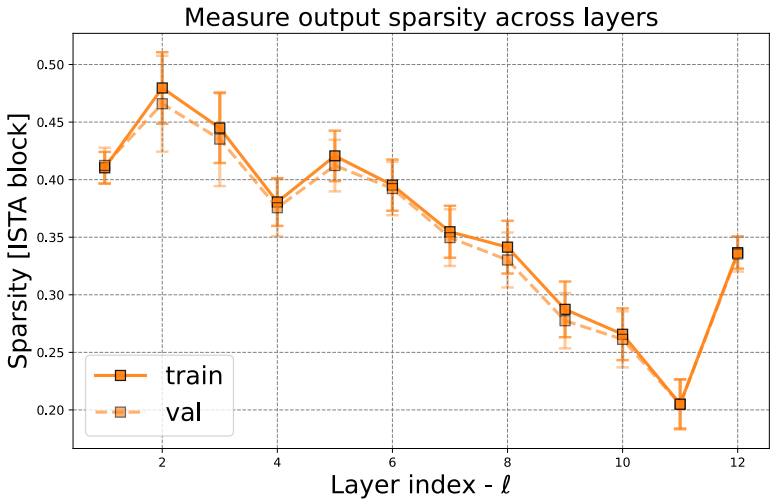
Datasets	CRATE-T	CRATE-S	CRATE-B	CRATE-L	ViT-T	ViT-S
# parameters	6.09M	13.12M	22.80M	77.64M	5.72M	22.05M
ImageNet	66.7	69.2	70.8	71.3	71.5	72.4
ImageNet ReaL	74.0	76.0	76.5	77.4	78.3	78.4
CIFAR10	95.5	96.0	96.8	97.2	96.6	97.2
CIFAR100	78.9	81.0	82.7	83.6	81.8	83.2
Oxford Flowers-102	84.6	87.1	88.7	88.3	85.1	88.5
Oxford-IIIT-Pets	81.4	84.9	85.3	87.4	88.5	88.6

- CRATE achieves performance close to thoroughly engineered vision transformers.
- Promising scaling behavior in CRATE.

Experiment II: Layer-wise Analysis of CRATE

Given a learned CRATE model, we measure the compression term of $\boldsymbol{Z}^{\ell+1/2}$ (*left*, $R^c(\boldsymbol{Z}^{\ell+1/2})$) and the sparsification term of $\boldsymbol{Z}^{\ell+1}$ (*right*, $\|\boldsymbol{Z}^{\ell+1}\|_0$) on train/validation samples at **each layer**.

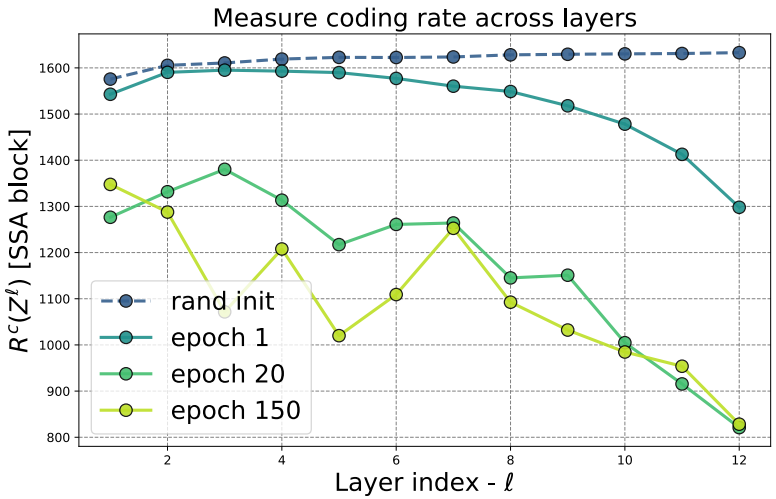


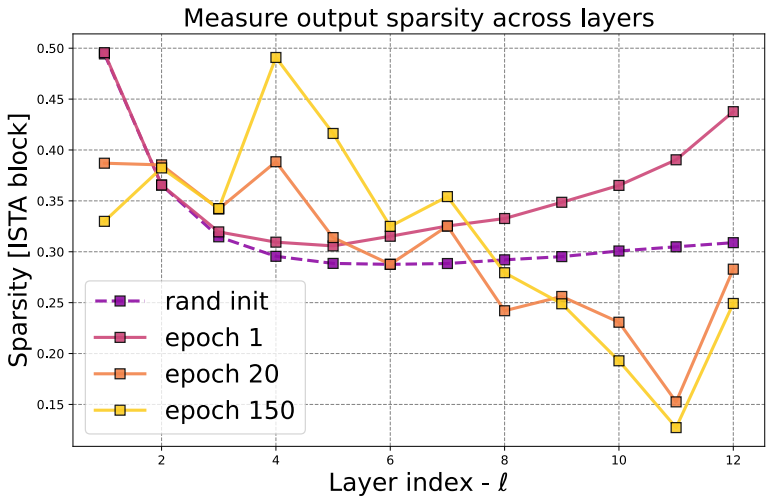


- The learned CRATE model indeed performs its design objective – each layer incrementally optimizes the compression term and the sparsification term.

Experiment II: Layer-wise Analysis of CRATE

For comparison, we measure the compression/sparsification term of randomly initialized CRATE model and models at different epochs.

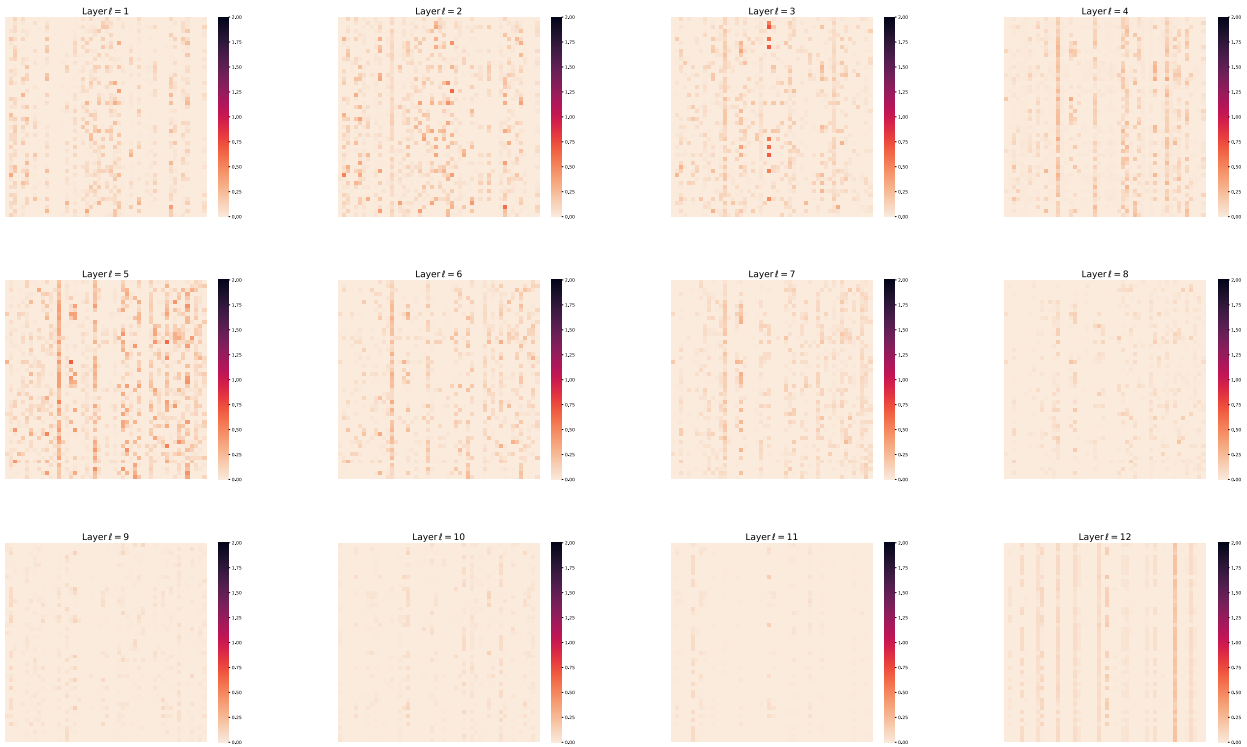




- Without learning from data, the random initialized CRATE model does not perform its design objective effectively.

Experiment III: Visualize Layer-wise Output of CRATE

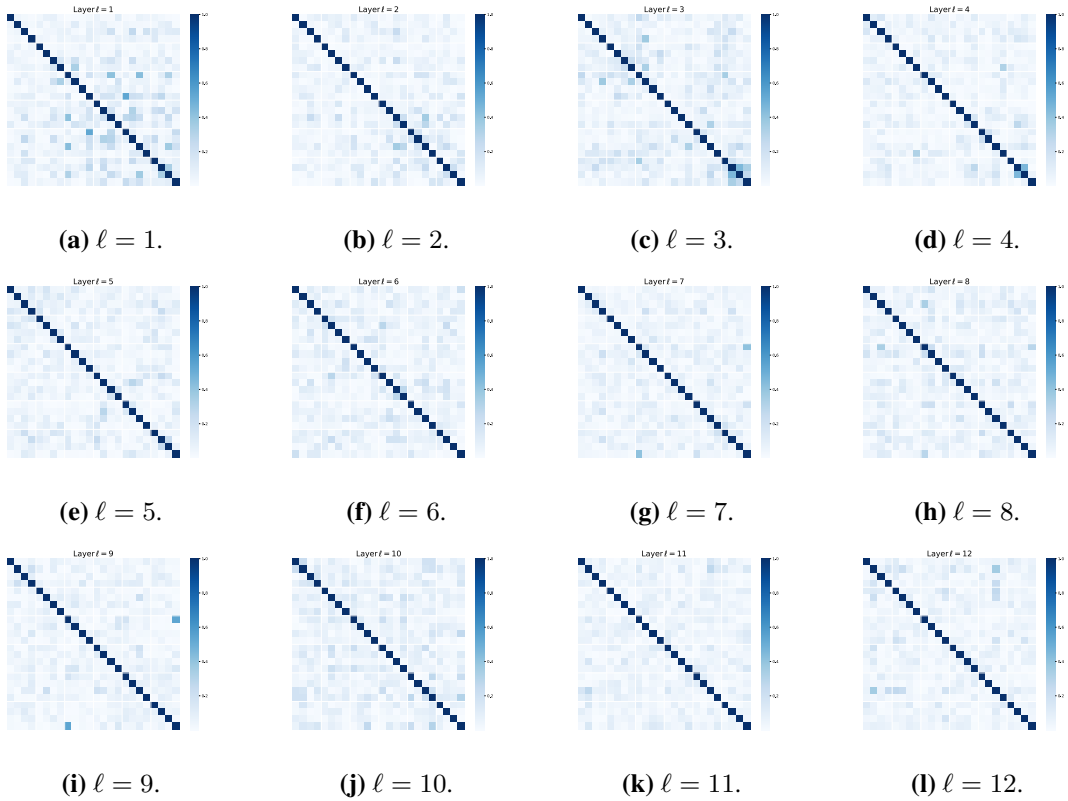
We use heatmaps to visualize the output of each layer in CRATE ($\boldsymbol{Z}^{\ell+1}$).



- We observe clear sparse and low-rank patterns of intermediate outputs of CRATE.

Experiment IV: Visualize Learned Subspaces of CRATE

We use heatmaps to visualize the correlations between different subspaces $(\boldsymbol{U}_k)_{k=1}^K$ of each `MSSA` layer in CRATE, i.e., $[\boldsymbol{U}_1^\ell, \ldots, \boldsymbol{U}_K^\ell]^* [\boldsymbol{U}_1^\ell, \ldots, \boldsymbol{U}_K^\ell]$.



(a) $\ell = 1$. **(b)** $\ell = 2$. **(c)** $\ell = 3$. **(d)** $\ell = 4$.

(e) $\ell = 5$. **(f)** $\ell = 6$. **(g)** $\ell = 7$. **(h)** $\ell = 8$.

(i) $\ell = 9$. **(j)** $\ell = 10$. **(k)** $\ell = 11$. **(l)** $\ell = 12$.

- The learned subspaces in `MSSA` blocks are incoherent.

Outline

Thank You! Questions?