Renormalization and Entropy as Principles for Representation Learning

Presenter: Francesco Caso

23rd October 2025

SAPIENZA
UNIVERSITÀ DI ROMA

RSTLESS

About me

Francesco Caso
PhD student, **Sapienza**
Member**, RSTLess**
Ex-Applied Scientist Intern, **Amazon**
Visiting PhD student**, Cambridge**

Supervised by

Fabrizio Silvestri
Full Professor, **Sapienza**
Head**, RSTLess**
Ex-Research Scientist, **Facebook AI**
Ex-Research Scientist, **Yahoo! Labs**

Pietro Liò
Full Professor, **Cambridge**
Member**, ELLIS**
Member, **AI Group**
Member**, CCAIM**

CONTACTS

SAPIENZA
UNIVERSITÀ DI ROMA

RSTLESS

How I got interested about it

Two words on Statistical Mechanics

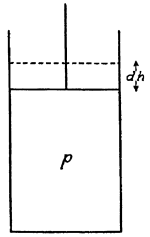
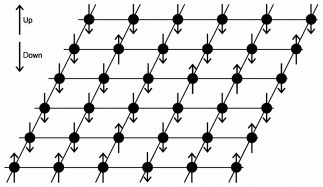


Fig. 1.



Entropy

Two words on Statistical Mechanics

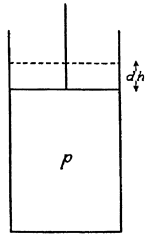
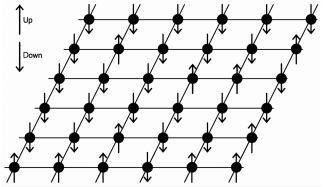
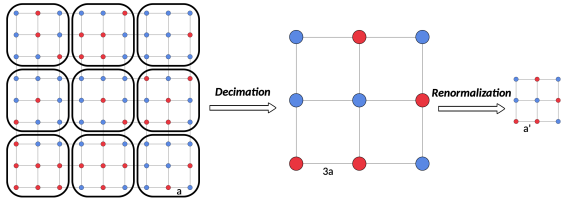
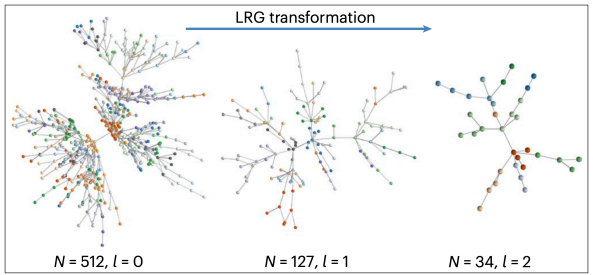


Fig. 1.



Entropy





Renormalization Group

Two words on Statistical Mechanics

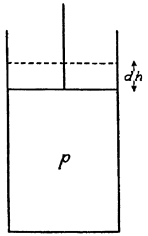
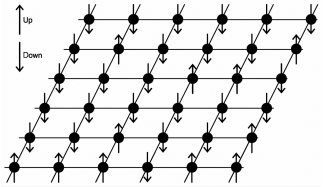
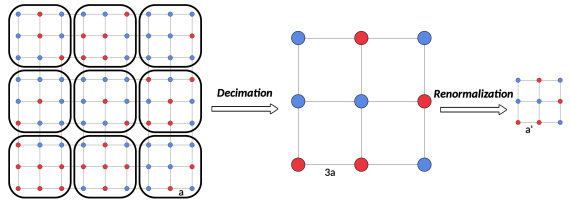
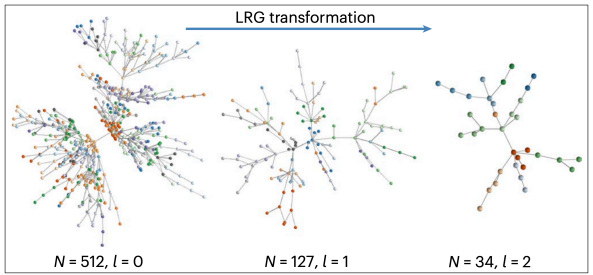


Fig. 1.

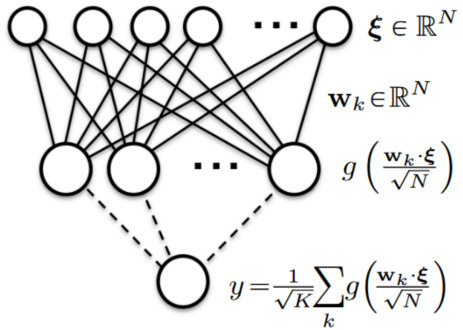


Entropy





Renormalization Group



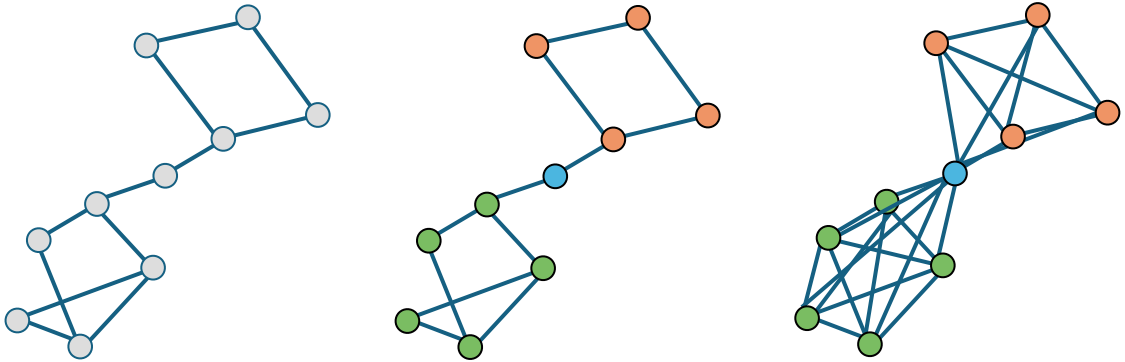
Representation learning

Imposing the RG

Renormalised Graph Representations for Node Classification

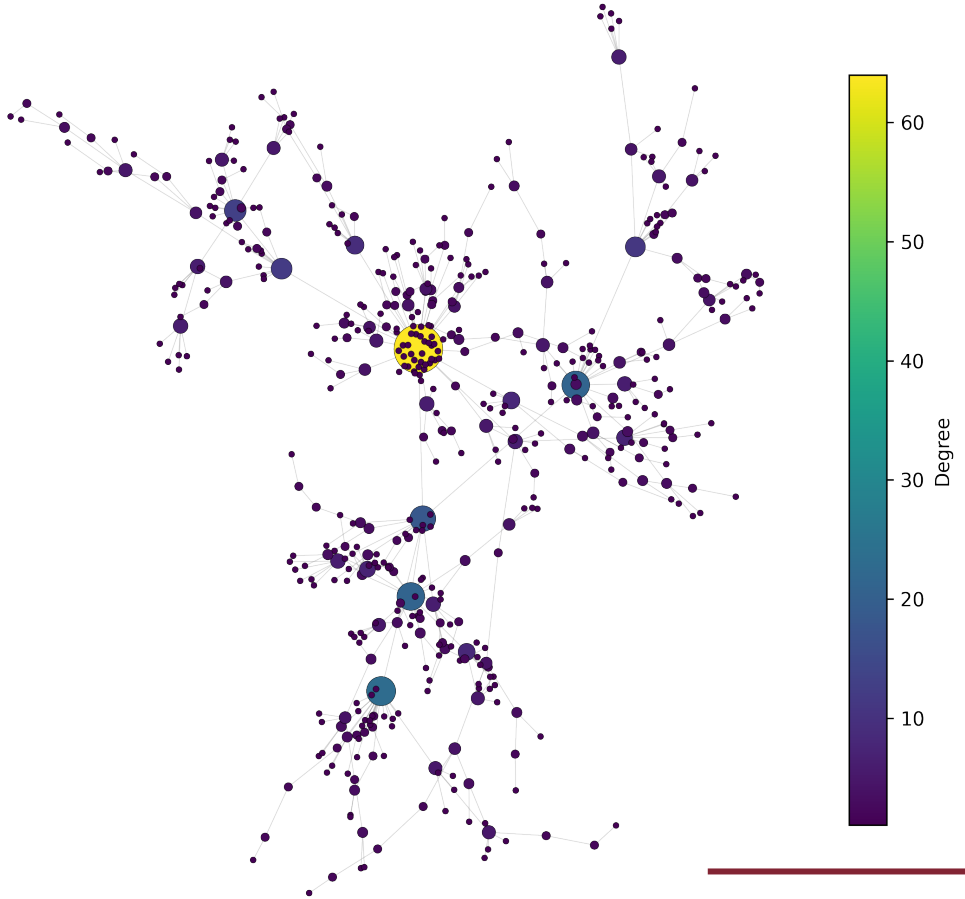
F. Caso, A. Bacciu, G. Trappolini, P. Liò, F. Silvestri

IJCNN 2025



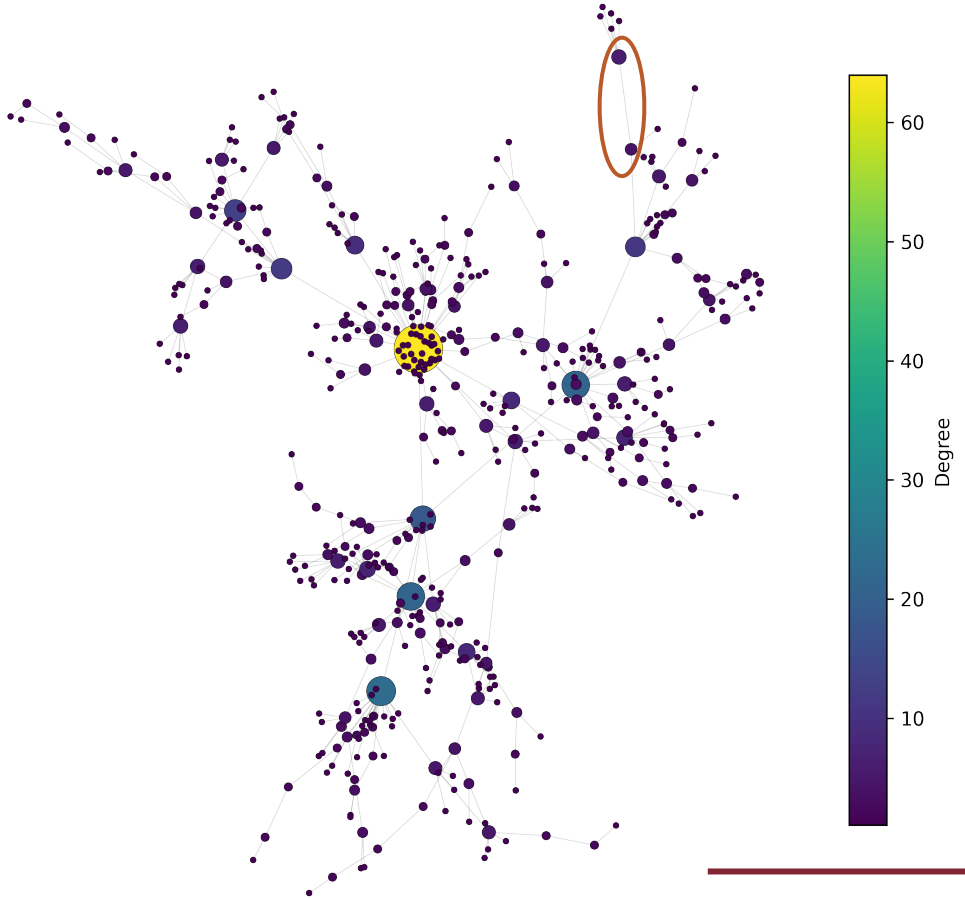
Motivation: Graphs encode structure at multiple levels

- Barabasi-Albert Graph with N=512 m=1



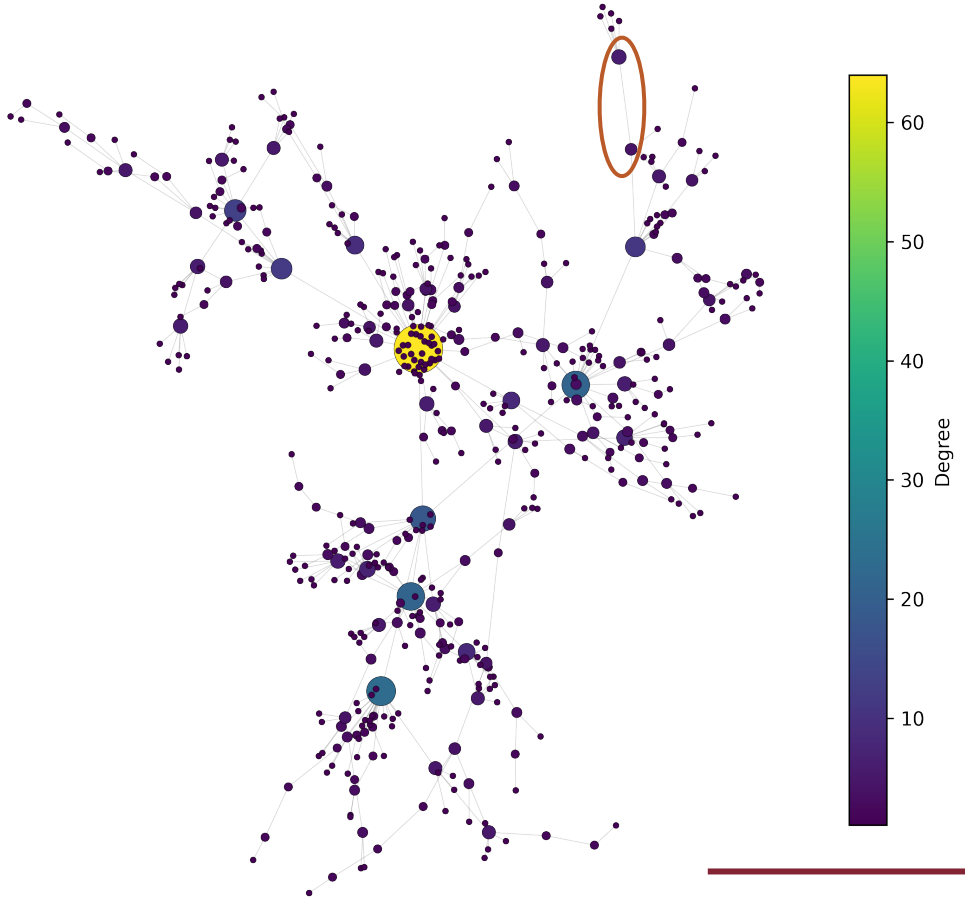
Motivation: Graphs encode structure at multiple levels

- Barabasi-Albert Graph with N=512 m=1
- Local Topology:
  - Neighbourhoods - edges



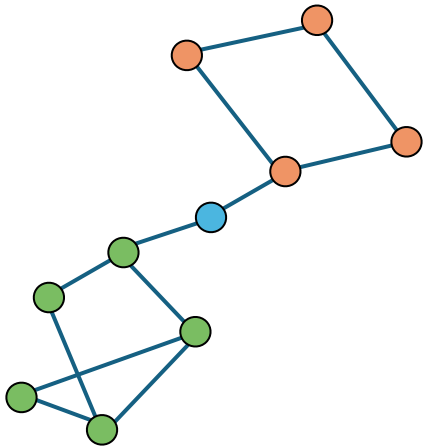
Motivation: Graphs encode structure at multiple levels

- Barabasi-Albert Graph with N=512 m=1
- Local Topology:
  - Neighbourhoods - edges
- Bigger Scale Topology:
  - Intersection of neighbourhoods - paths



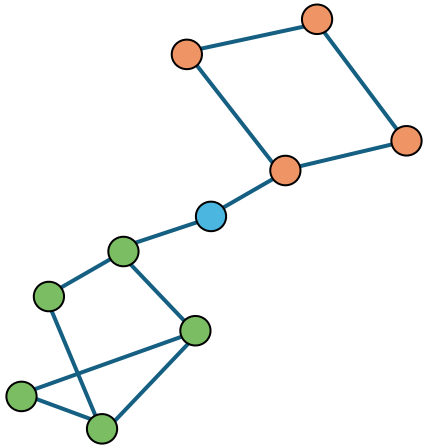
Going Deeper Is Not the Same as Seeing Further

Original

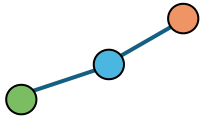


Going Deeper Is Not the Same as Seeing Further

Original



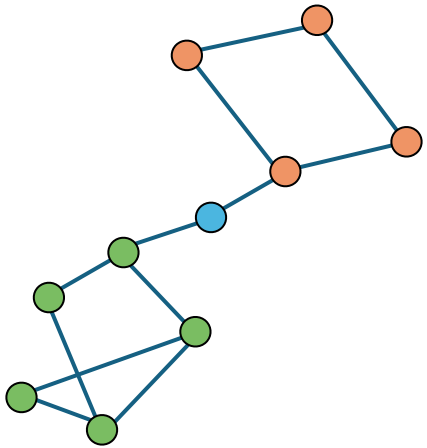
Shallow



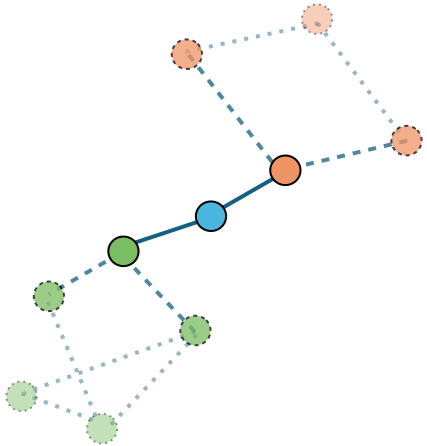
Going Deeper Is Not the Same as Seeing Further

- **Mesoscopic** patterns are not reconstructed by just adding layers

Original



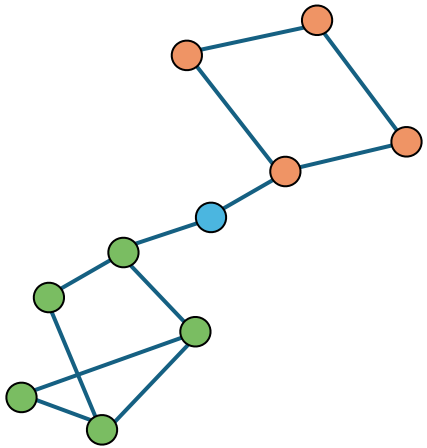
Deep



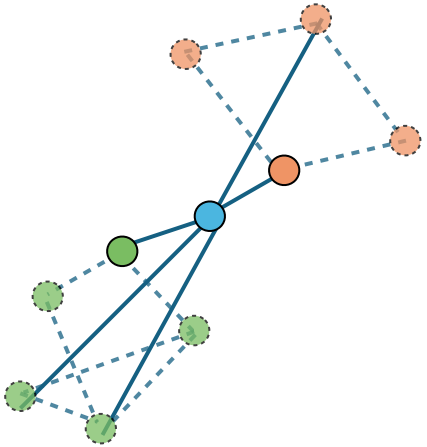
Going Deeper Is Not the Same as Seeing Further

- **Rewiring** surface hidden topological signals by altering/erasing some patterns.

Original



Rewired + Deep

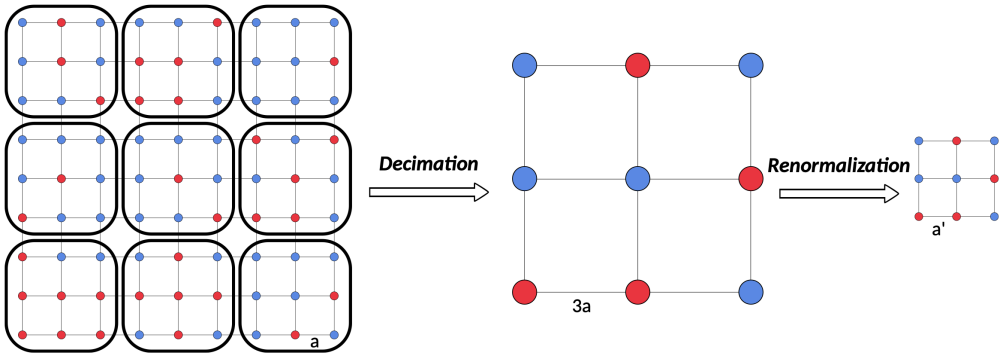


Can we define an **approach that rewires the graph based on scale?**

Selectively discarding fine-grained details, preserving coarser structure

Renormalization Group (RG): Looking at Graphs from Different Scales

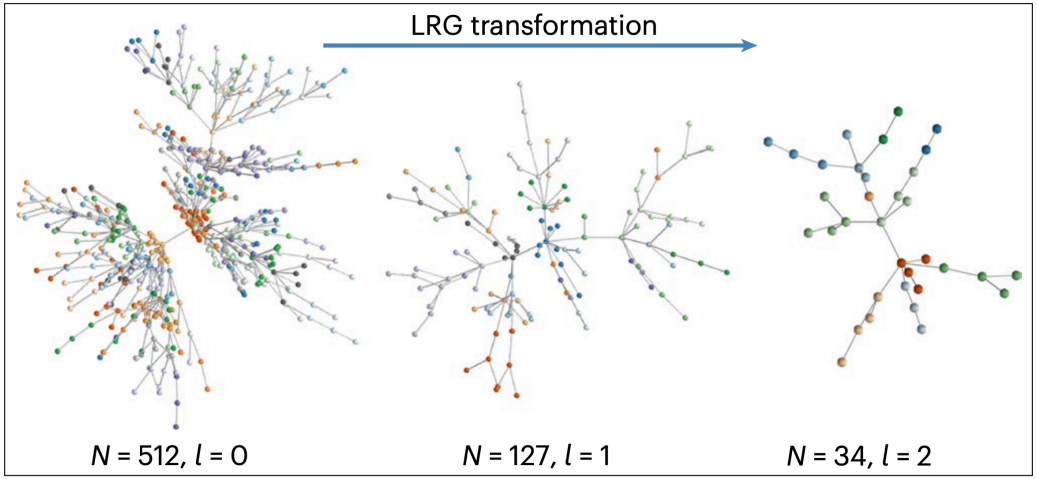
- **RG**: how a model should change when we change the scale.
- In **Euclidean space**, scale is intuitive.



Villegas et al. 2023

Laplacian Renormalization Group (LRG): Looking at Graphs from Different Scales

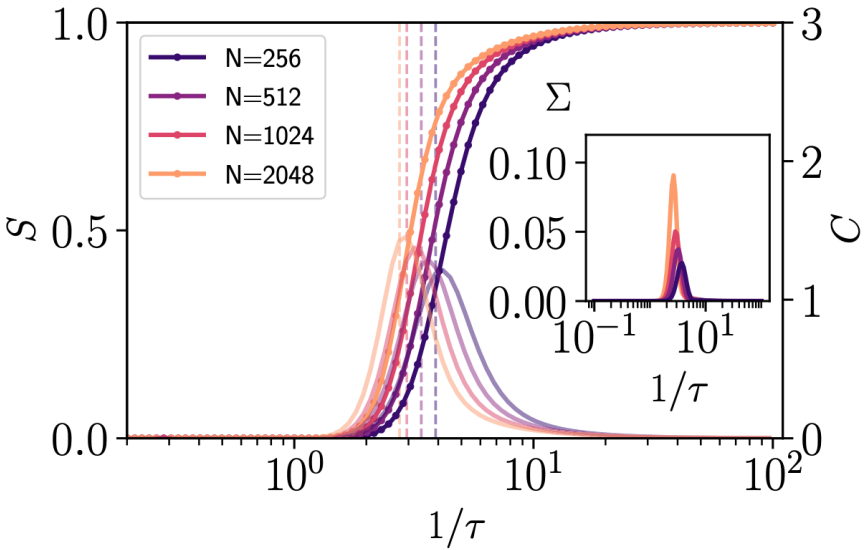
- In **graphs**, scale is not as intuitive.
- Instead of redefining scale, **we modify the spatial operators**.
- **The Laplacian**, that defines diffusion.



Villegas et al. 2023

Diffusion and Entropy on Graphs

- $L_{ij} = [(\delta_{ij} \sum_k A_{ik}) - A_{ij}]$
- $\rho(\tau) = \frac{e^{-\tau L}}{Tr(e^{-\tau L})}$
- $v(\tau) = \rho(\tau) v(0)$
- $S[\rho(\tau)] = -\frac{1}{\log(N)} \sum_{i=1}^{N} \mu_i(\tau) \log \mu_i(\tau)$

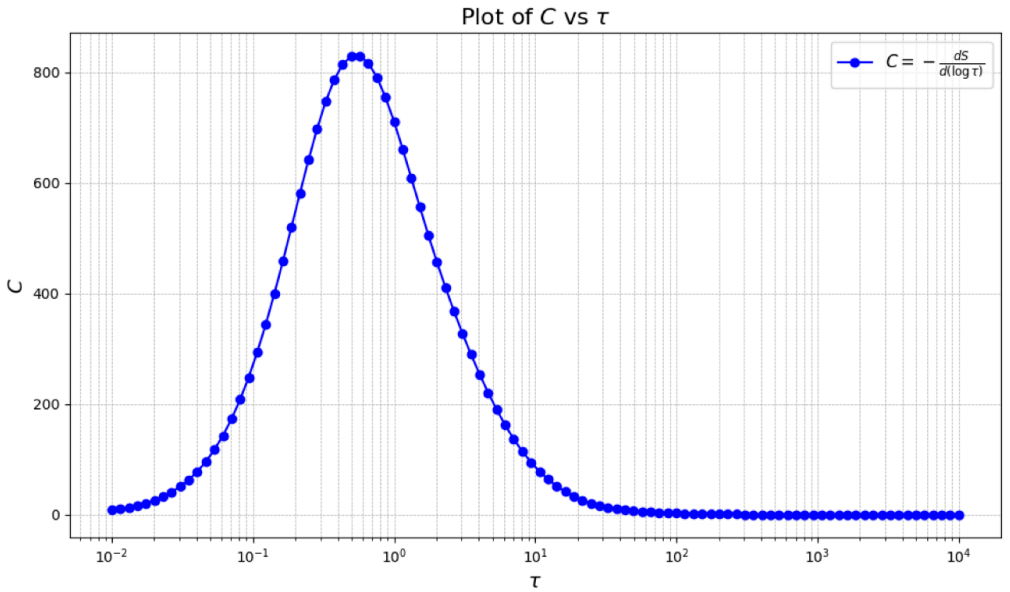


Villegas et al. 2022

Characteristic Scale: A Theoretically-Grounded Choice

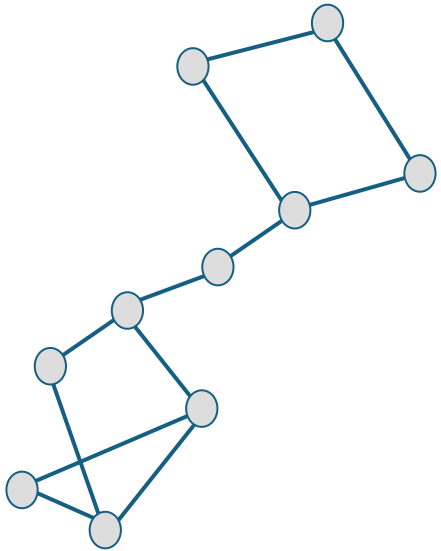
- $$C = -\frac{dS}{d(\log \tau)}$$

- The **peak** in the **entropy's derivative** (heat capacity) reveals the **characteristic scale**, representing strong intra-cluster coupling.



Our Framework: Rewiring the Graph

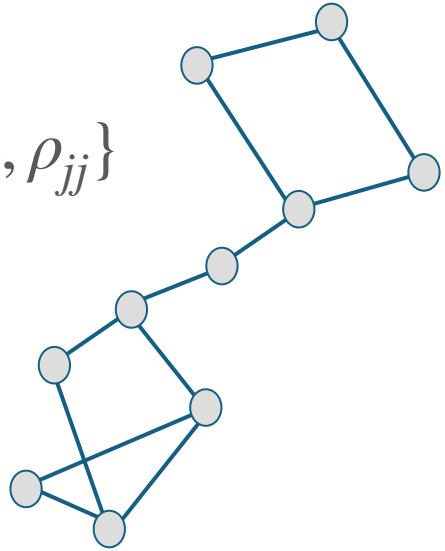
Original



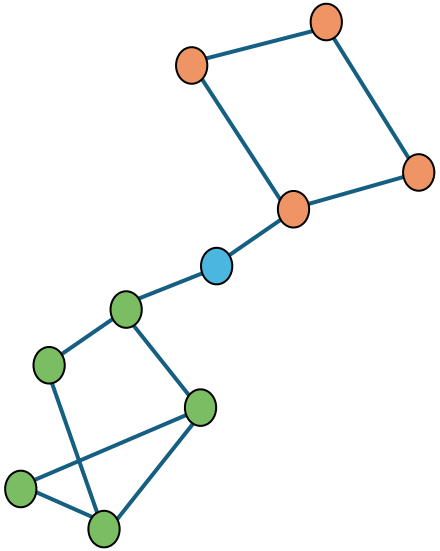
Our Framework: Rewiring the Graph

- **Condition:** $\rho_{i,j}(\tau) > \max\{\rho_{ii}, \rho_{jj}\}$
- **Then:** $\mathcal{N}_{i'} = \mathcal{N}_i \cup \mathcal{N}_j$

Original

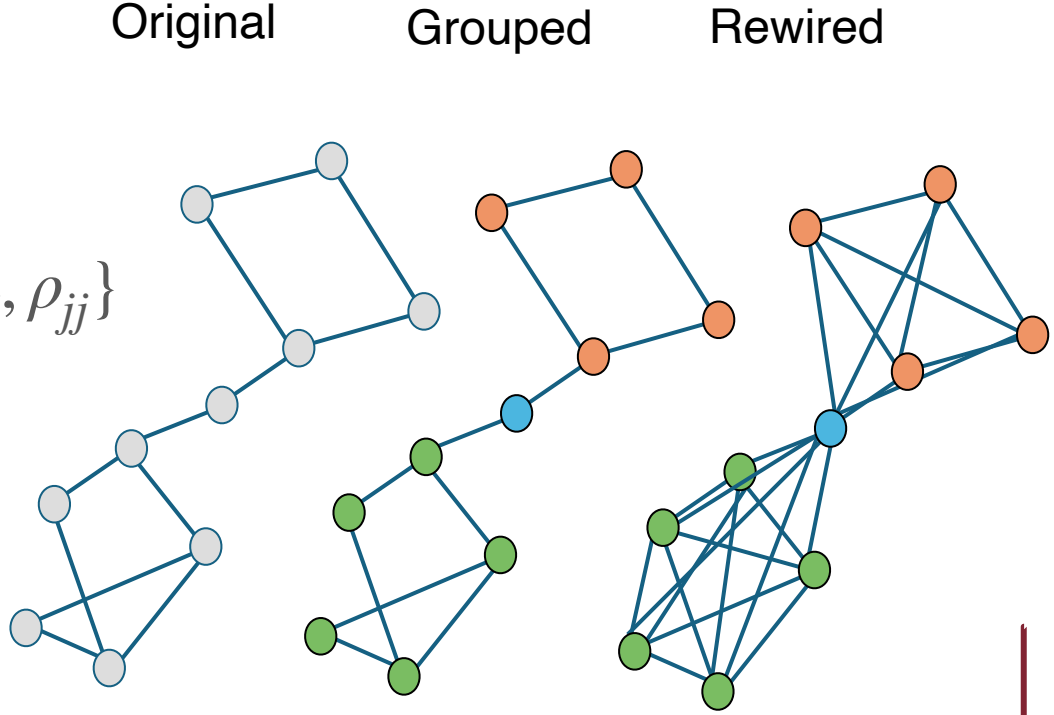


Grouped



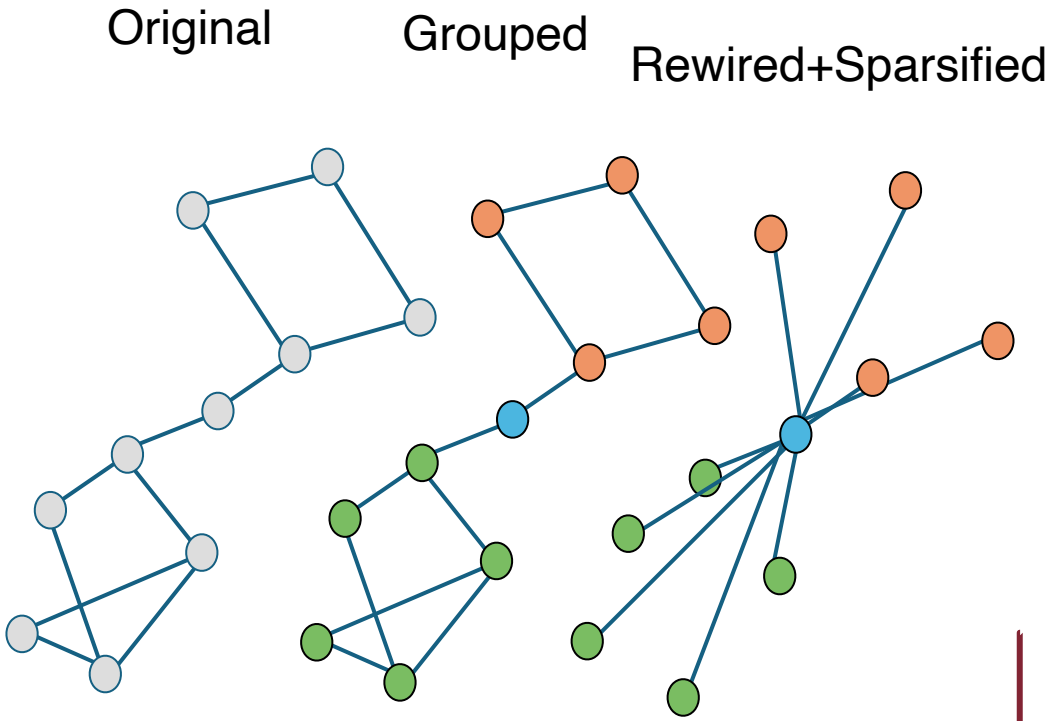
Our Framework: Rewiring the Graph

- **Condition:** $\rho_{i,j}(\tau) > \max\{\rho_{ii}, \rho_{jj}\}$
- **Then:** $\mathcal{N}_{i'} = \mathcal{N}_i \cup \mathcal{N}_j$



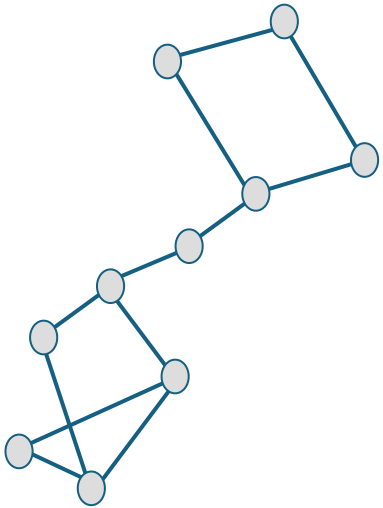
Our Framework: Rewiring the Graph

- **Intra-macro-node sparsification**

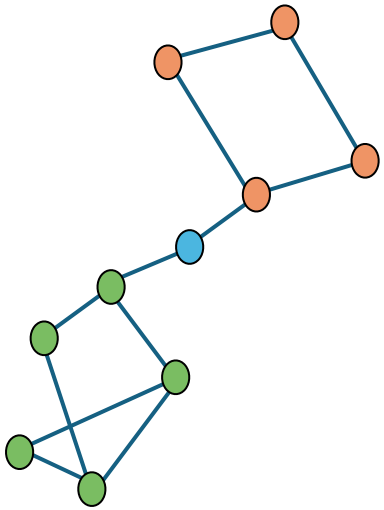


Our Framework: Rewiring the Graph

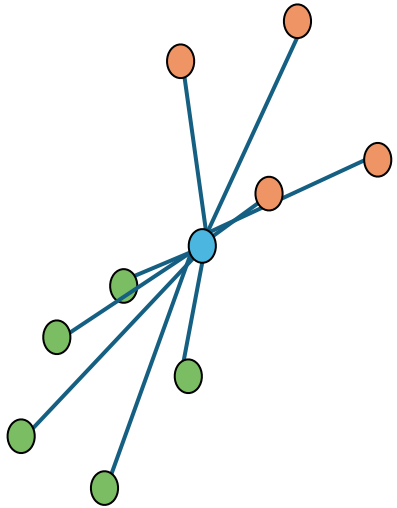
Original



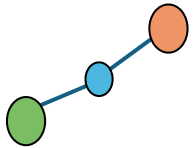
Grouped



Rewired + Sparsified



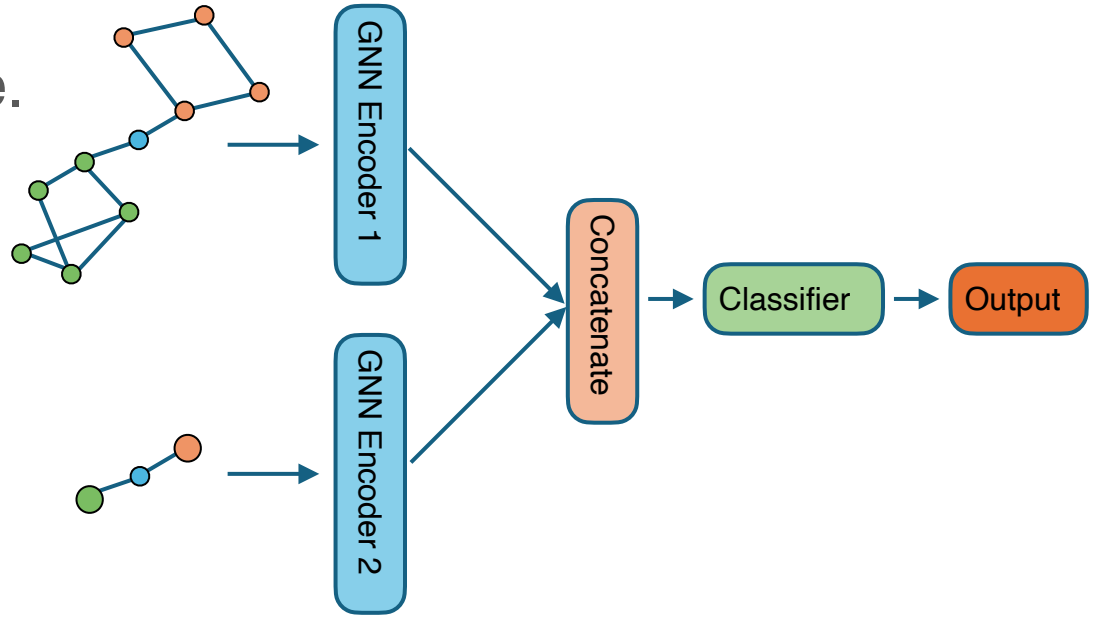
Coarse



Our Framework: Processing Multiple Graph Scales in Parallel

- Encoders can be GCN, GAT, or any architecture.

- The model learns to **combine local and mesoscopic information**.



Is it beneficial to observe multiple scales of a graph for performing a node classification task?

Experimental Setup: Datasets and Training Configuration

- Datasets (from citation, air traffic, and product networks)
- We propagate only on the train subgraph during training!

Dataset	Nodes	Edges	Features	Classes
Citeseer	3,327	4,732	3,703	6
Cora	2,708	5,429	1,433	7
Europe	399	5,995	399	4
PubMed	19,717	44,338	500	3
Photo	238162	7650	745	8
Computers	491722	13752	767	10

Results: Multiscale Models Achieve Higher Accuracy

- **SB** = Single Base
- **MB** = Multi Base (more encoders, only original graph)
- **MR** = Multi Renormalised (encoders on graph at different resolutions)
- **+** = statistically significant improvement (Wilcoxon test)

Model	Citeseer	Cora	Europe	Pubmed	Photo	Computers
Base						
Single						
GAT_{SB}	65.6±2.0	72.6±1.7	34.2±5.7	71.1±1.5	95.9±0.2	88.0±1.5
GCN_{SB}	65.8±1.6	72.0±0.7	35.1±5.3	73.9±1.5	96.0±0.2	92.0±0.3
Multi						
GAT_{MB}	66.3±2.4	73.6±1.4	30.8±4.6	71.9±1.2	96.1±0.3	91.5±0.4
GCN_{MB}	65.7±2.7	72.4±1.1	35.8±5.1	73.9±1.0	96.2±0.2	92.6±0.2
Renormalized						
Single						
GAT_{SR}	$67.5\pm1.7^{+}$	$74.2\pm0.9^{+}$	$30.1\pm7.7^{-}$	$72.2\pm1.1^{+}$	$92.1\pm0.7^{-}$	$73.0\pm4.8^{-}$
GCN_{SR}	$69.4\pm1.5^{+}$	72.0±1.0	$24.4\pm7.0^{-}$	73.5±1.5	$90.3\pm0.6^{-}$	$74.7\pm3.2^{-}$
Multi						
GAT_{MR}	$69.0\pm1.7^{+}$	$75.5\pm1.2^{+}$	29.9±4.5	$73.6\pm0.8^{+}$	96.2±0.4	$89.7\pm1.0^{-}$
GCN_{MR}	$69.0\pm2.5^{+}$	$75.2\pm1.3^{+}$	38.6±4.6	$76.1\pm1.2^{+}$	96.3±0.2	$91.9\pm0.3^{-}$

- Red: best
- Orange: second best

Results: Multiscale Models Achieve Higher Accuracy

- **SB** = Single Base
- **MB** = Multi Base (more encoders, only original graph)
- **MR** = Multi Renormalised (encoders on graph at different resolutions)
- **+** = statistically significant improvement (Wilcoxon test)

- Red: best
- Orange: second best

Model	Citeseer	Cora	Europe	Pubmed	Photo	Computers
Base						
Single						
GAT_{SB}	65.6±2.0	72.6±1.7	34.2±5.7	71.1±1.5	95.9±0.2	88.0±1.5
GCN_{SB}	65.8±1.6	72.0±0.7	35.1±5.3	73.9±1.5	96.0±0.2	92.0±0.3
Multi						
GAT_{MB}	66.3±2.4	73.6±1.4	30.8±4.6	71.9±1.2	96.1±0.3	91.5±0.4
GCN_{MB}	65.7±2.7	72.4±1.1	35.8±5.1	73.9±1.0	96.2±0.2	92.6±0.2
Renormalized						
Single						
GAT_{SR}	$67.5±1.7^{+}$	$74.2±0.9^{+}$	$30.1±7.7^{-}$	$72.2±1.1^{+}$	$92.1±0.7^{-}$	$73.0±4.8^{-}$
GCN_{SR}	$69.4±1.5^{+}$	72.0±1.0	$24.4±7.0^{-}$	73.5±1.5	$90.3±0.6^{-}$	$74.7±3.2^{-}$
Multi						
GAT_{MR}	$69.0±1.7^{+}$	$75.5±1.2^{+}$	29.9±4.5	$73.6±0.8^{+}$	96.2±0.4	$89.7±1.0^{-}$
GCN_{MR}	$69.0±2.5^{+}$	$75.2±1.3^{+}$	38.6±4.6	$76.1±1.2^{+}$	96.3±0.2	$91.9±0.3^{-}$

Results: Multiscale Models Achieve Higher Accuracy

- **SB** = Single Base
- **MB** = Multi Base (more encoders, only original graph)
- **MR** = Multi Renormalised (encoders on graph at different resolutions)
- **+** = statistically significant improvement (Wilcoxon test)

- Red: best
- Orange: second best

Model	Citeseer	Cora	Europe	Pubmed	Photo	Computers
Base						
Single						
GAT_{SB}	65.6±2.0	72.6±1.7	34.2±5.7	71.1±1.5	95.9±0.2	88.0±1.5
GCN_{SB}	65.8±1.6	72.0±0.7	35.1±5.3	73.9±1.5	96.0±0.2	92.0±0.3
Multi						
GAT_{MB}	66.3±2.4	73.6±1.4	30.8±4.6	71.9±1.2	96.1±0.3	91.5±0.4
GCN_{MB}	65.7±2.7	72.4±1.1	35.8±5.1	73.9±1.0	96.2±0.2	92.6±0.2
Renormalized						
Single						
GAT_{SR}	$67.5±1.7^{+}$	$74.2±0.9^{+}$	$30.1±7.7^{-}$	$72.2±1.1^{+}$	$92.1±0.7^{-}$	$73.0±4.8^{-}$
GCN_{SR}	$69.4±1.5^{+}$	72.0±1.0	$24.4±7.0^{-}$	73.5±1.5	$90.3±0.6^{-}$	$74.7±3.2^{-}$
Multi						
GAT_{MR}	$69.0±1.7^{+}$	$75.5±1.2^{+}$	29.9±4.5	$73.6±0.8^{+}$	96.2±0.4	$89.7±1.0^{-}$
GCN_{MR}	$69.0±2.5^{+}$	$75.2±1.3^{+}$	38.6±4.6	$76.1±1.2^{+}$	96.3±0.2	$91.9±0.3^{-}$

Results: Multiscale Models Achieve Higher Accuracy

- **SB** = Single Base
- **MB** = Multi Base (more encoders, only original graph)
- **MR** = Multi Renormalised (encoders on graph at different resolutions)
- **+** = statistically significant improvement (Wilcoxon test)

Model	Citeseer	Cora	Europe	Pubmed	Photo	Computers
Base						
Single						
GAT_{SB}	65.6±2.0	72.6±1.7	34.2±5.7	71.1±1.5	95.9±0.2	88.0±1.5
GCN_{SB}	65.8±1.6	72.0±0.7	35.1±5.3	73.9±1.5	96.0±0.2	92.0±0.3
Multi						
GAT_{MB}	66.3±2.4	73.6±1.4	30.8±4.6	71.9±1.2	96.1±0.3	91.5±0.4
GCN_{MB}	65.7±2.7	72.4±1.1	35.8±5.1	73.9±1.0	96.2±0.2	92.6±0.2
Renormalized						
Single						
GAT_{SR}	$67.5\pm1.7^{+}$	$74.2\pm0.9^{+}$	$30.1\pm7.7^{-}$	$72.2\pm1.1^{+}$	$92.1\pm0.7^{-}$	$73.0\pm4.8^{-}$
GCN_{SR}	$69.4\pm1.5^{+}$	72.0±1.0	$24.4\pm7.0^{-}$	73.5±1.5	$90.3\pm0.6^{-}$	$74.7\pm3.2^{-}$
Multi						
GAT_{MR}	$69.0\pm1.7^{+}$	$75.5\pm1.2^{+}$	29.9±4.5	$73.6\pm0.8^{+}$	96.2±0.4	$89.7\pm1.0^{-}$
GCN_{MR}	$69.0\pm2.5^{+}$	$75.2\pm1.3^{+}$	38.6±4.6	$76.1\pm1.2^{+}$	96.3±0.2	$91.9\pm0.3^{-}$

- Red: best
- Orange: second best

Results: Multiscale Models Achieve Higher Accuracy

- **SB** = Single Base
- **MB** = Multi Base (more encoders, only original graph)
- **MR** = Multi Renormalised (encoders on graph at different resolutions)
- **+** = statistically significant improvement (Wilcoxon test)

Model	Citeseer	Cora	Europe	Pubmed	Photo	Computers
Base						
Single						
GAT_{SB}	65.6±2.0	72.6±1.7	34.2±5.7	71.1±1.5	95.9±0.2	88.0±1.5
GCN_{SB}	65.8±1.6	72.0±0.7	35.1±5.3	73.9±1.5	96.0±0.2	92.0±0.3
Multi						
GAT_{MB}	66.3±2.4	73.6±1.4	30.8±4.6	71.9±1.2	96.1±0.3	91.5±0.4
GCN_{MB}	65.7±2.7	72.4±1.1	35.8±5.1	73.9±1.0	96.2±0.2	92.6±0.2
Renormalized						
Single						
GAT_{SR}	67.5±1.7^{+}	74.2±0.9^{+}	30.1±7.7^{-}	72.2±1.1^{+}	92.1±0.7^{-}	73.0±4.8^{-}
GCN_{SR}	69.4±1.5^{+}	72.0±1.0	24.4±7.0^{-}	73.5±1.5	90.3±0.6^{-}	74.7±3.2^{-}
Multi						
GAT_{MR}	69.0±1.7^{+}	75.5±1.2^{+}	29.9±4.5	73.6±0.8^{+}	96.2±0.4	89.7±1.0^{-}
GCN_{MR}	69.0±2.5^{+}	75.2±1.3^{+}	38.6±4.6	76.1±1.2^{+}	96.3±0.2	91.9±0.3^{-}

- Red: best
- Orange: second best

Results: Multiscale Models Achieve Higher Accuracy

- **SB** = Single Base
- **MB** = Multi Base (more encoders, only original graph)
- **MR** = Multi Renormalised (encoders on graph at different resolutions)
- **+** = statistically significant improvement (Wilcoxon test)

- Red: best
- Orange: second best

Model	Citeseer	Cora	Europe	Pubmed	Photo	Computers
Base						
Single						
GAT_{SB}	65.6±2.0	72.6±1.7	34.2±5.7	71.1±1.5	95.9±0.2	88.0±1.5
GCN_{SB}	65.8±1.6	72.0±0.7	35.1±5.3	73.9±1.5	96.0±0.2	92.0±0.3
Multi						
GAT_{MB}	66.3±2.4	73.6±1.4	30.8±4.6	71.9±1.2	96.1±0.3	91.5±0.4
GCN_{MB}	65.7±2.7	72.4±1.1	35.8±5.1	73.9±1.0	96.2±0.2	92.6±0.2
Renormalized						
Single						
GAT_{SR}	$67.5±1.7^{+}$	$74.2±0.9^{+}$	$30.1±7.7^{-}$	$72.2±1.1^{+}$	$92.1±0.7^{-}$	$73.0±4.8^{-}$
GCN_{SR}	$69.4±1.5^{+}$	72.0±1.0	$24.4±7.0^{-}$	73.5±1.5	$90.3±0.6^{-}$	$74.7±3.2^{-}$
Multi						
GAT_{MR}	$69.0±1.7^{+}$	$75.5±1.2^{+}$	29.9±4.5	$73.6±0.8^{+}$	96.2±0.4	$89.7±1.0^{-}$
GCN_{MR}	$69.0±2.5^{+}$	$75.2±1.3^{+}$	38.6±4.6	$76.1±1.2^{+}$	96.3±0.2	$91.9±0.3^{-}$

Results: Multiscale Models Achieve Higher Accuracy

- **SB** = Single Base
- **MB** = Multi Base (more encoders, only original graph)
- **MR** = Multi Renormalised (encoders on graph at different resolutions)
- **+** = statistically significant improvement (Wilcoxon test)

Model	Citeseer	Cora	Europe	Pubmed	Photo	Computers
Base						
Single						
GAT_{SB}	65.6±2.0	72.6±1.7	34.2±5.7	71.1±1.5	95.9±0.2	88.0±1.5
GCN_{SB}	65.8±1.6	72.0±0.7	35.1±5.3	73.9±1.5	96.0±0.2	92.0±0.3
Multi						
GAT_{MB}	66.3±2.4	73.6±1.4	30.8±4.6	71.9±1.2	96.1±0.3	91.5±0.4
GCN_{MB}	65.7±2.7	72.4±1.1	35.8±5.1	73.9±1.0	96.2±0.2	92.6±0.2
Renormalized						
Single						
GAT_{SR}	$67.5\pm1.7^{+}$	$74.2\pm0.9^{+}$	$30.1\pm7.7^{-}$	$72.2\pm1.1^{+}$	$92.1\pm0.7^{-}$	$73.0\pm4.8^{-}$
GCN_{SR}	$69.4\pm1.5^{+}$	72.0±1.0	$24.4\pm7.0^{-}$	73.5±1.5	$90.3\pm0.6^{-}$	$74.7\pm3.2^{-}$
Multi						
GAT_{MR}	$69.0\pm1.7^{+}$	$75.5\pm1.2^{+}$	29.9±4.5	$73.6\pm0.8^{+}$	96.2±0.4	$89.7\pm1.0^{-}$
GCN_{MR}	$69.0\pm2.5^{+}$	$75.2\pm1.3^{+}$	38.6±4.6	$76.1\pm1.2^{+}$	96.3±0.2	$91.9\pm0.3^{-}$

- Red: best
- Orange: second best

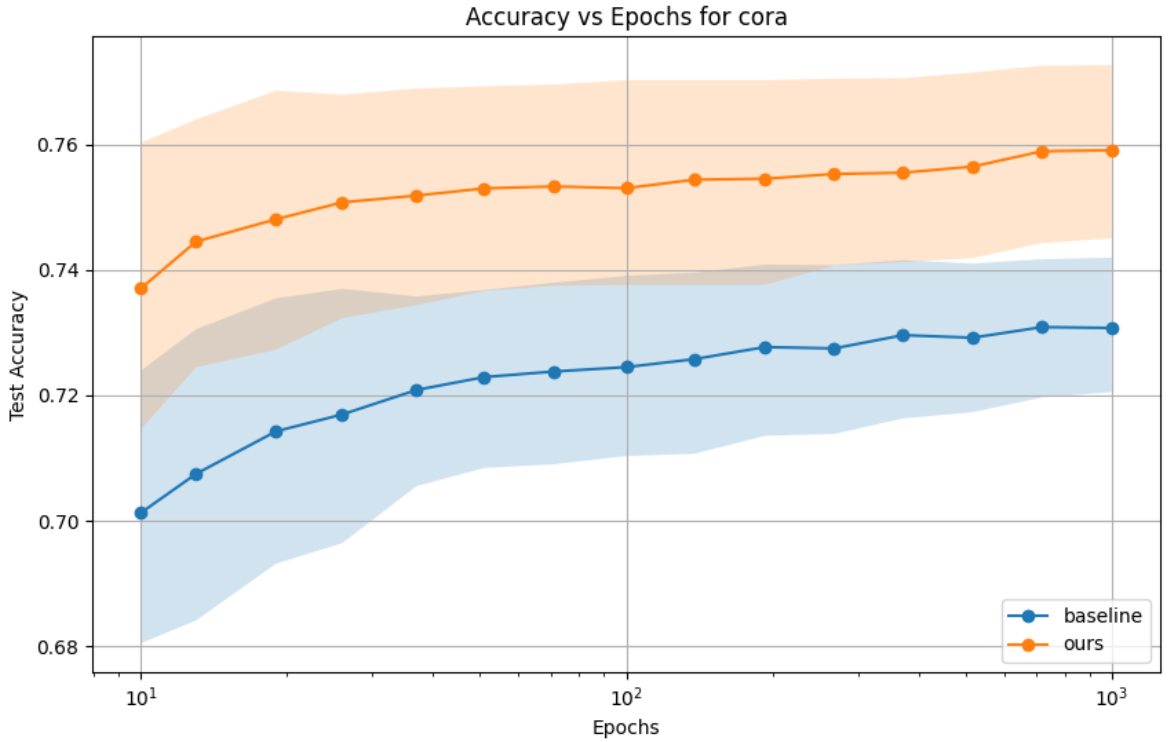
Results: Multiscale Models Achieve Higher Accuracy

- **SB** = Single Base
- **MB** = Multi Base (more encoders, only original graph)
- **MR** = Multi Renormalised (encoders on graph at different resolutions)
- **+** = statistically significant improvement (Wilcoxon test)

Model	Citeseer	Cora	Europe	Pubmed	Photo	Computers
Base						
Single						
GAT_{SB}	65.6±2.0	72.6±1.7	34.2±5.7	71.1±1.5	95.9±0.2	88.0±1.5
GCN_{SB}	65.8±1.6	72.0±0.7	35.1±5.3	73.9±1.5	96.0±0.2	92.0±0.3
Multi						
GAT_{MB}	66.3±2.4	73.6±1.4	30.8±4.6	71.9±1.2	96.1±0.3	91.5±0.4
GCN_{MB}	65.7±2.7	72.4±1.1	35.8±5.1	73.9±1.0	96.2±0.2	92.6±0.2
Renormalized						
Single						
GAT_{SR}	$67.5±1.7^{+}$	$74.2±0.9^{+}$	$30.1±7.7^{-}$	$72.2±1.1^{+}$	$92.1±0.7^{-}$	$73.0±4.8^{-}$
GCN_{SR}	$69.4±1.5^{+}$	72.0±1.0	$24.4±7.0^{-}$	73.5±1.5	$90.3±0.6^{-}$	$74.7±3.2^{-}$
Multi						
GAT_{MR}	$69.0±1.7^{+}$	$75.5±1.2^{+}$	29.9±4.5	$73.6±0.8^{+}$	96.2±0.4	$89.7±1.0^{-}$
GCN_{MR}	$69.0±2.5^{+}$	$75.2±1.3^{+}$	38.6±4.6	$76.1±1.2^{+}$	96.3±0.2	$91.9±0.3^{-}$

- Red: best
- Orange: second best

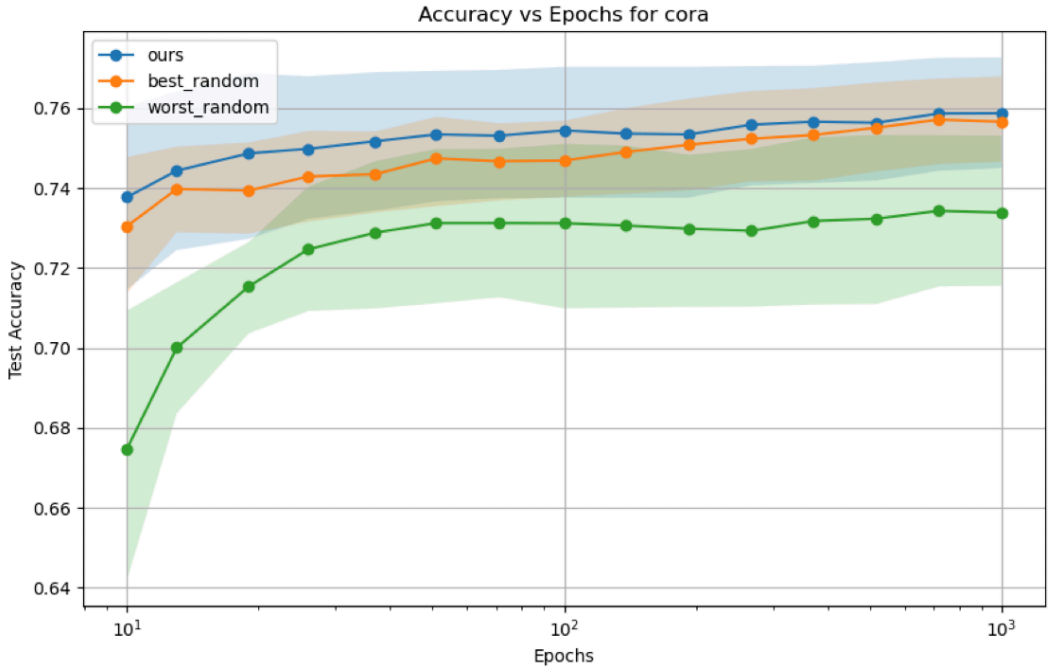
Multiscale Advantage Is Consistent During Training



Can we systematically identify the optimal scales using spectral entropy?

Our Scale Is Optimal: No Tuning Required

- Characteristic scale derived from spectral entropy.
- Outperforms all randomly chosen scales (30 tested across 3 ranges).
- No tuning, no cross-validation: selected **before training**.



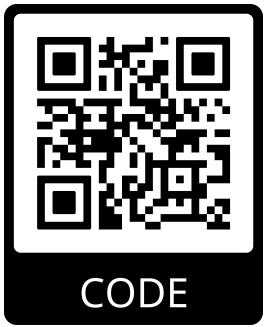
Limitations and Future Directions

- Limitations
- LRG applies only to **undirected, unweighted, single-component graphs**.
- Does **not consider edge features** or **node features** during scale selection.
- On large dense graphs (e.g. **Amazon Computers**), performance **may degrade**.

- Future Work
- Extend RG methods to **directed** or **weighted** graphs.
- Define **feature-aware spectral entropies** for task-specific scaling.
- Apply to **graph classification** or **link prediction**.

Conclusions…of this project

- Can we build a bridge between **rewiring** and **rescaling**? **Yes**
- Is it beneficial to observe **multiple scales** of a graph for performing a **node classification** task? **Yes**
- Can we **systematically** identify the **optimal scales** using spectral entropy? **Yes**
- We need to include **features** in the definition of **graph entropy**.



Learning the RG

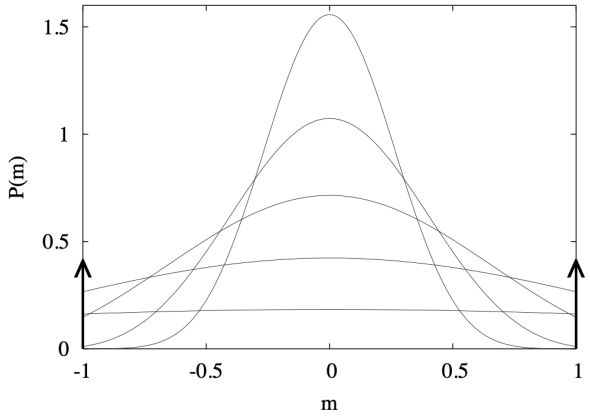
Symmetry and Generalisation in Neural Approximations of Renormalisation Transformations

C. Ashworth, P. Liò, **F. Caso**

Preprint on arXiv

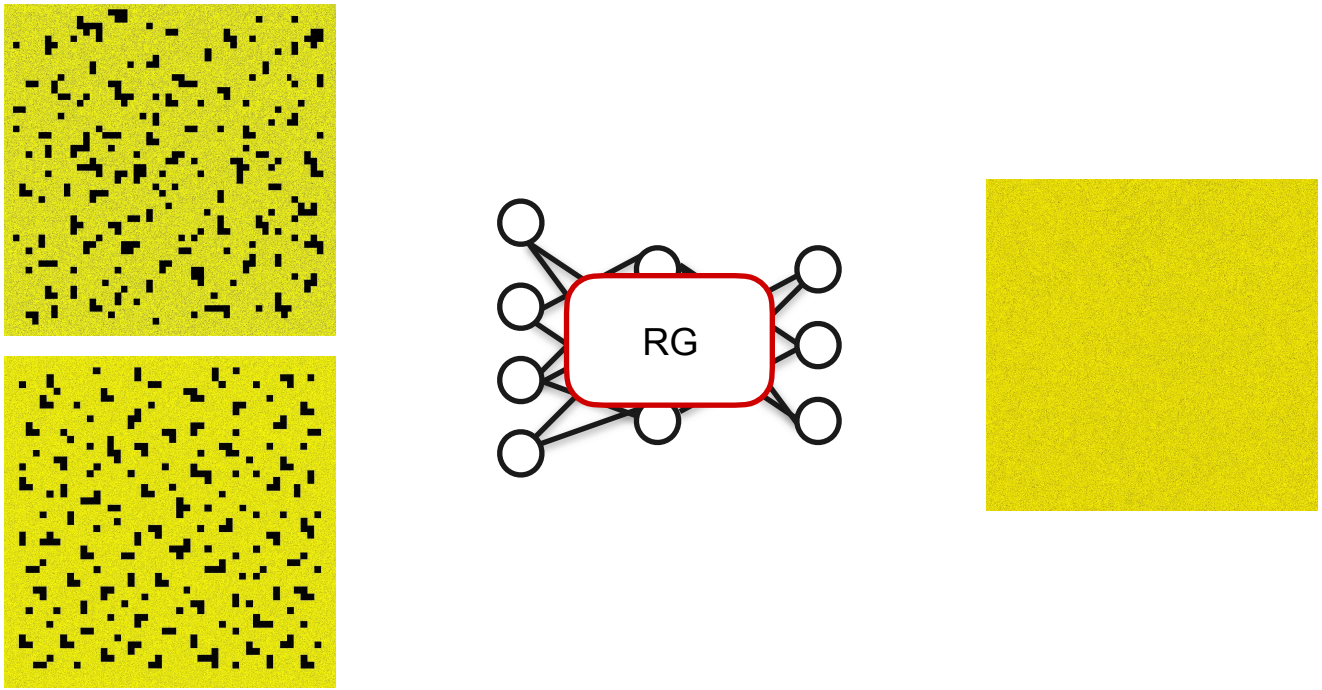
$$\mu_{y^l,i} = \quad + \quad + \quad = \quad \mu_{z^l,i} \quad + \alpha\,(\mu_{z^l,i})^2 \quad + \alpha\,\Sigma_{z^l,ii}\,,$$

[Fischer et al.]

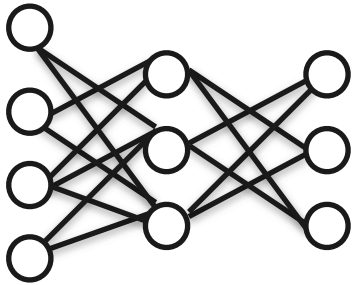


[Antal et al.]

Motivation: Are Neural Networks RG flows?



Motivation: Are Neural Networks RG flows?



Cat

Motivation: Learning RG flows?

- In the pedagogical case of a ϕ^4 theory, RG flows reduces to few equations evolving few variables.

Motivation: Learning RG flows?

- In the pedagogical case of a ϕ^4 theory, RG flows reduces to few equations evolving few variables.
- In real cases (e.g. fermionic systems) RG is computationally difficult ($O(10^5)$ equations) but has proven useful.

Motivation: Learning RG flows?

- In the pedagogical case of a ϕ^4 theory, RG flows reduces to few equations evolving few variables.
- In real cases (e.g. fermionic systems) RG is computationally difficult ($O(10^5)$ equations) but has proven useful.
- E.g. the Hubbard model represents cuprates and organic superconductors

$$H = -t \sum_{\langle i,j \rangle, s} c^{\dagger}_{i,s} c_{j,s} - t' \sum_{\langle\langle i,j \rangle\rangle, s} c^{\dagger}_{i,s} c_{j,s} + U \sum_{i} n_{i,\uparrow} n_{i,\downarrow}$$

Motivation: Learning RG flows?

E.g. the Hubbard model represents cuprates and organic superconductors

$$H = -t \sum_{\langle i,j \rangle, s} c^{\dagger}_{i,s} c_{j,s} - t' \sum_{\langle\langle i,j \rangle\rangle, s} c^{\dagger}_{i,s} c_{j,s} + U \sum_{i} n_{i,\uparrow} n_{i,\downarrow}$$

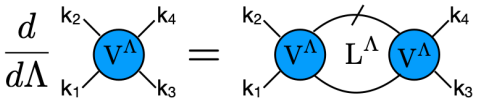
Di Sante et al.

Motivation: Learning RG flows?

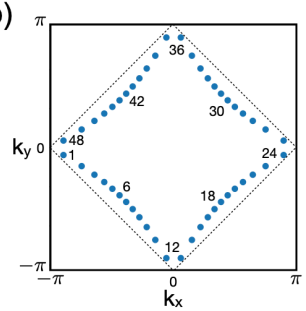
E.g. the Hubbard model represents cuprates and organic superconductors

$$H = -t \sum_{\langle i,j \rangle, s} c^{\dagger}_{i,s} c_{j,s} - t' \sum_{\langle\langle i,j \rangle\rangle, s} c^{\dagger}_{i,s} c_{j,s} + U \sum_{i} n_{i,\uparrow} n_{i,\downarrow}$$

a)



b)



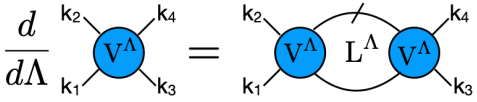
Di Sante et al.

Motivation: Learning RG flows?

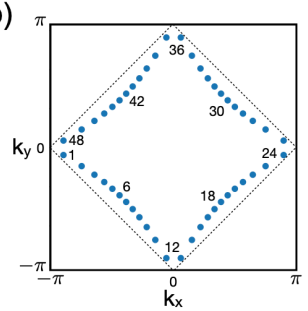
E.g. the Hubbard model represents cuprates and organic superconductors

$$H = -t \sum_{\langle i,j \rangle, s} c^{\dagger}_{i,s} c_{j,s} - t' \sum_{\langle\langle i,j \rangle\rangle, s} c^{\dagger}_{i,s} c_{j,s} + U \sum_{i} n_{i,\uparrow} n_{i,\downarrow}$$

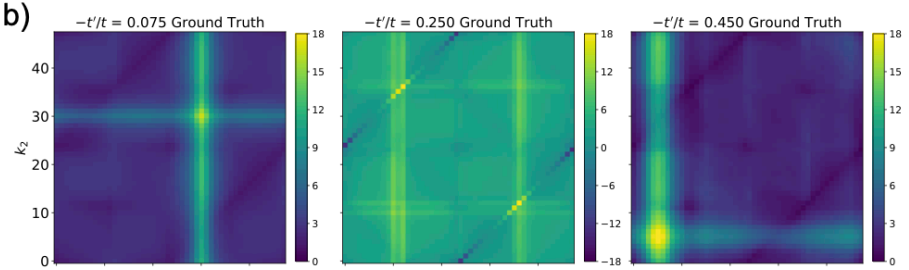
a)



b)



b)



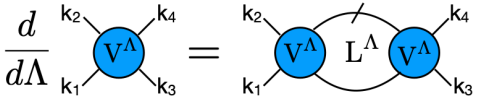
Di Sante et al.

Motivation: Learning RG flows?

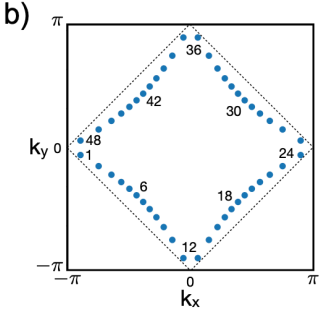
E.g. the Hubbard model represents cuprates and organic superconductors

$$H = -t \sum_{\langle i,j \rangle, s} c^{\dagger}_{i,s} c_{j,s} - t' \sum_{\langle\langle i,j \rangle\rangle, s} c^{\dagger}_{i,s} c_{j,s} + U \sum n_{i,\uparrow} n_{i,\downarrow}$$

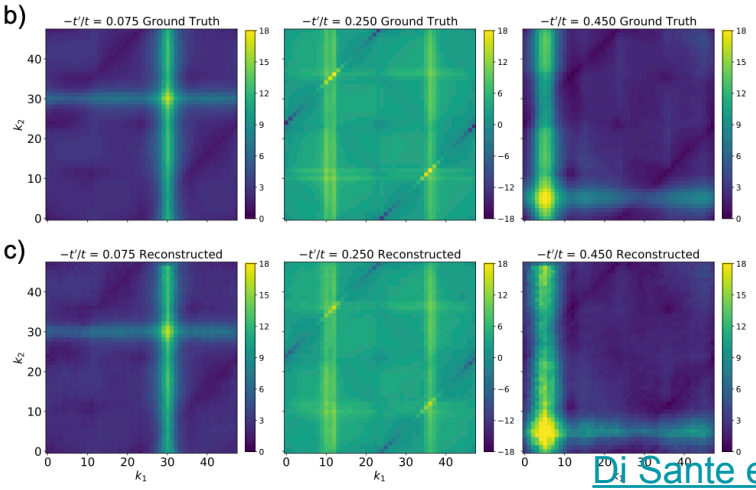
a)



b)

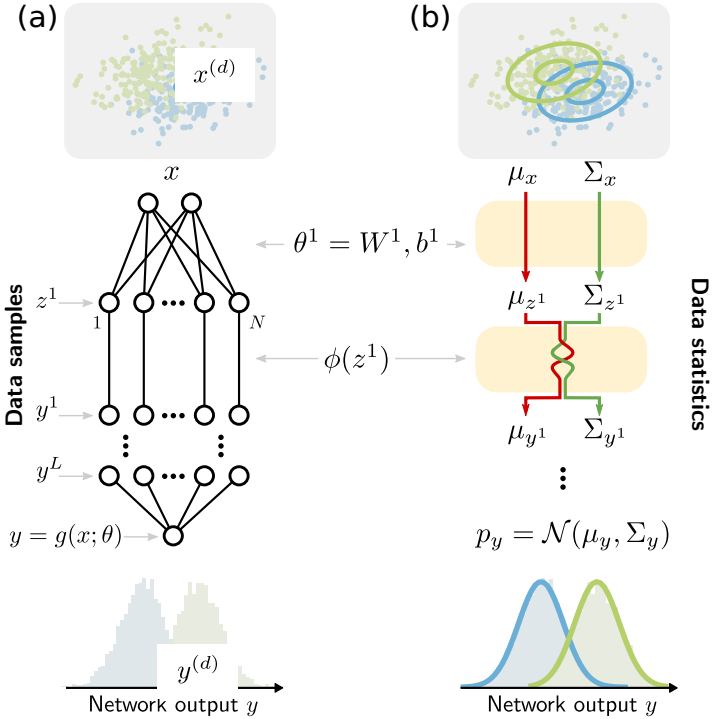


b)



Di Sante et al.

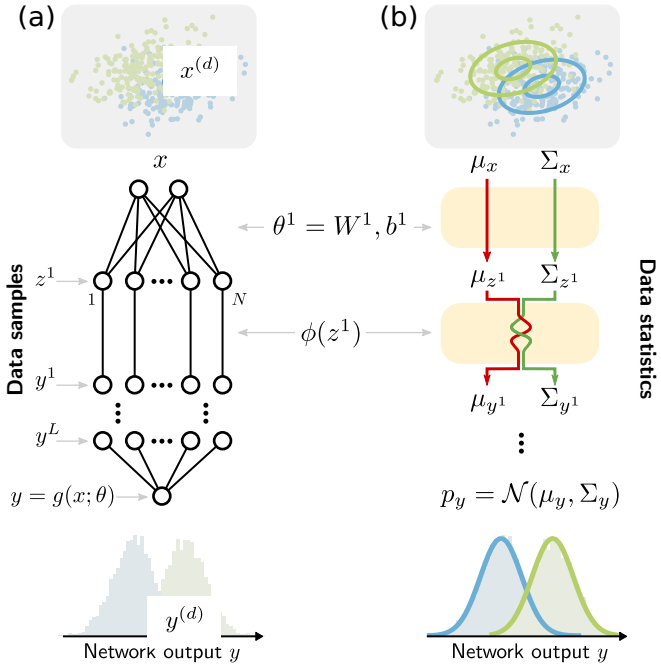
Background: NNs as mapping Cumulants



Background: NNs as mapping Cumulants

Moments: $< x >, < x^2 >, \ldots$

Cumulants: $< x >, < x^2 > - (< x >)^2, \ldots$



Fischer et al.

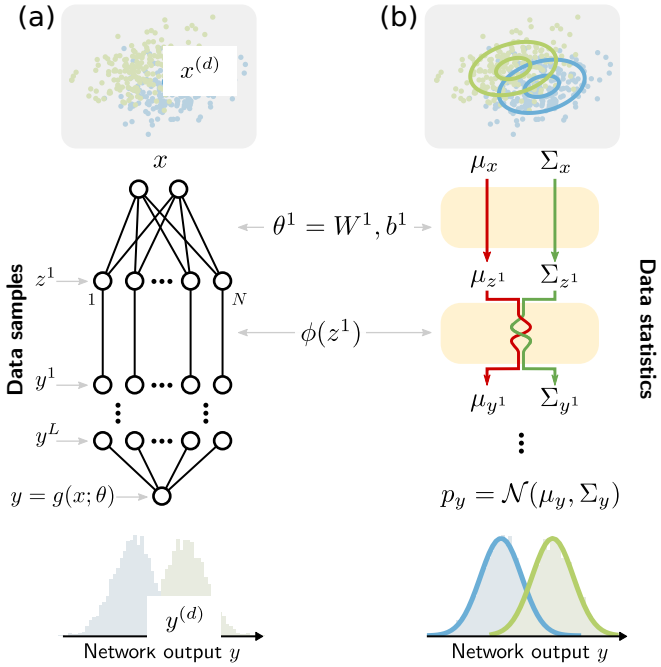
Background: NNs as mapping Cumulants

Moments: $< x >, < x^2 >, \ldots$

Cumulants: $< x >, < x^2 > - (< x >)^2, \ldots$

We can define the cumulant generating function:

$$\mathscr{W}_{y|\theta}(j) = \ln\langle \exp(j^\top y)\rangle_{y|\theta}$$
$$= \ln\langle \exp(j^\top g(x;\theta))\rangle_x,$$



Background: NNs as mapping Cumulants

Moments: $< x >, < x^2 >, \ldots$

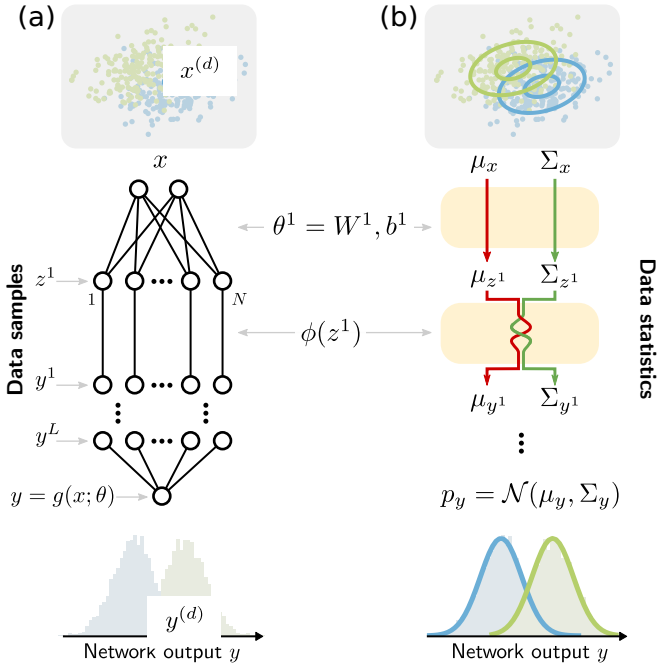
Cumulants: $< x >, < x^2 > - (< x >)^2, \ldots$

We can define the cumulant generating function:

$$\mathscr{W}_{y|\theta}(j) = \ln\langle \exp(j^\top y)\rangle_{y|\theta}$$
$$= \ln\langle \exp(j^\top g(x;\theta))\rangle_x,$$

Which…generates cumulants:

$$k^{(n)}_{y|\theta} = \left.\frac{d^n \mathscr{W}_{y|\theta}(j)}{dj^n}\right|_{j=0}.$$



Background: NNs as mapping Cumulants

An MLP is composed by an affine transform:

$$z_i^l = \sum_{j=1}^{N_{l-1}} W_{ij}^l \, y_j^{l-1} + b_i^l$$

Background: NNs as mapping Cumulants

An MLP is composed by an affine transform:

$$z_i^l = \sum_{j=1}^{N_{l-1}} W_{ij}^l \, y_j^{l-1} + b_i^l$$

And an activation function:

$$y_i^l = \phi\left(z_i^l\right) = \phi\left(\sum_{j=1}^{N_{l-1}} W_{ij}^l \, y_j^{l-1} + b_i^l\right).$$

Background: NNs as mapping Cumulants

An MLP is composed by an affine transform: $z_i^l = \sum_{j=1}^{N_{l-1}} W_{ij}^l \, y_j^{l-1} + b_i^l;$

Background: NNs as mapping Cumulants

An MLP is composed by an affine transform: $z_i^l = \sum_{j=1}^{N_{l-1}} W_{ij}^l \, y_j^{l-1} + b_i^l;$

$$\begin{aligned}
\mathscr{W}_{z^l}(j) &= \ln\langle \exp(j^\top z^l)\rangle_{z^l} \\
&= \ln\langle \exp(j^\top W^l y^{l-1} + j^\top b^l)\rangle_{y^{l-1}} \quad ; \\
&= \mathscr{W}_{y^{l-1}}((W^l)^\top j) + j^\top b^l;
\end{aligned}$$

Background: NNs as mapping Cumulants

An MLP is composed by an affine transform: $z_i^l = \sum_{j=1}^{N_{l-1}} W_{ij}^l \, y_j^{l-1} + b_i^l;$

$$
\begin{aligned}
\mathscr{W}_{z^l}(j) &= \ln\langle \exp(j^\top z^l) \rangle_{z^l} \\
&= \ln\langle \exp(j^\top W^l y^{l-1} + j^\top b^l) \rangle_{y^{l-1}} \quad ; \; k^{(n)}_{z^l, i_1, \ldots, i_n} = \sum_{s_1, \ldots, s_n} W^l_{i_1 s_1} \cdots W^l_{i_n s_n} \, k^{(n)}_{y^{l-1}, s_1, \ldots, s_n} \\
&= \mathscr{W}_{y^{l-1}}((W^l)^\top j) + j^\top b^l;
\end{aligned}
$$

Background: NNs as mapping Cumulants

An MLP is composed by an affine transform: $z_i^l = \sum_{j=1}^{N_{l-1}} W_{ij}^l \, y_j^{l-1} + b_i^l;$

$$\begin{aligned}\mathscr{W}_{z^l}(j) &= \ln\langle \exp(j^\top z^l)\rangle_{z^l} \\ &= \ln\langle \exp(j^\top W^l y^{l-1} + j^\top b^l)\rangle_{y^{l-1}} \\ &= \mathscr{W}_{y^{l-1}}((W^l)^\top j) + j^\top b^l;\end{aligned}$$

$$; \; k^{(n)}_{z^l,i_1,\ldots,i_n} = \sum_{s_1,\ldots,s_n} W^l_{i_1 s_1} \cdots W^l_{i_n s_n} \, k^{(n)}_{y^{l-1},s_1,\ldots,s_n}$$

Cumulant don't mix because of linearity

Background: NNs as mapping Cumulants

[…] And an activation function:

$$y_i^l = \phi\left(z_i^l\right) = \phi\left(\sum_{j=1}^{N_{l-1}} W_{ij}^l\, y_j^{l-1} + b_i^l\right).$$

Background: NNs as mapping Cumulants

[…] And an activation function:

$$y_i^l = \phi\left(z_i^l\right) = \phi\left(\sum_{j=1}^{N_{l-1}} W_{ij}^l\, y_j^{l-1} + b_i^l\right).$$

$$\begin{aligned}\mathscr{W}_{y^l}(j) &= \ln\langle \exp(j^\top y^l)\rangle_{y^l} \\ &= \ln\langle \exp(j^\top \phi(z^l))\rangle_{z^l}\,.\end{aligned}$$

Background: NNs as mapping Cumulants

[…] And an activation function:

$$y_i^l = \phi\left(z_i^l\right) = \phi\left(\sum_{j=1}^{N_{l-1}} W_{ij}^l\, y_j^{l-1} + b_i^l\right).$$

$$\begin{aligned}\mathscr{W}_{y^l}(j) &= \ln\langle \exp(j^\top y^l)\rangle_{y^l} \\ &= \ln\langle \exp(j^\top \phi(z^l))\rangle_{z^l}\,.\end{aligned}$$

Cumulant do mix and
we need to approximate it

Background: NNs as mapping Cumulants

[…] And an activation function:

$$y_i^l = \phi\left(z_i^l\right) = \phi\left(\sum_{j=1}^{N_{l-1}} W_{ij}^l\, y_j^{l-1} + b_i^l\right).$$

$$\begin{aligned}\mathscr{W}_{y^l}(j) &= \ln\langle \exp(j^\top y^l)\rangle_{y^l} \\ &= \ln\langle \exp(j^\top \phi(z^l))\rangle_{z^l}\,.\end{aligned}$$

$$\mu_{y^l,i} = \quad + \quad + \quad$$

$$= \mu_{z^l,i} + \alpha\,(\mu_{z^l,i})^2 + \alpha\,\Sigma_{z^l,ii}\,,$$

Backgorund: RG as mapping Probability distributions

CLT: Let $\xi_1, \ldots, \xi_n, \ldots$ i.i.d. random variables from a distribution with variance σ^2

Then, $$\frac{\sum_i (\xi_i - \mathbb{E}(\xi_i))}{\sigma\sqrt{n}} \xrightarrow{n\to\infty} N(0,1)\,.$$

Backgorund: RG as mapping Probability distributions

CLT: Let $\xi_1, \ldots, \xi_n, \ldots$ i.i.d. random variables from a distribution with variance σ^2

Then, $$\frac{\sum_i (\xi_i - \mathbb{E}(\xi_i))}{\sigma\sqrt{n}} \xrightarrow{n\to\infty} N(0,1)\,.$$

Backgorund: RG as mapping Probability distributions

CLT: Let $\xi_1, \ldots, \xi_n, \ldots$ i.i.d. random variables from a distribution with variance σ^2

Then, $$\frac{\sum_i (\xi_i - \mathbb{E}(\xi_i))}{\sigma\sqrt{n}} \xrightarrow{n\to\infty} N(0,1)\,.$$

As an RG:

$$p_{n+1}(x) = \sqrt{2}\int dy\, p_n(\sqrt{2}x - y)\, p_n(y) = (\mathscr{R}p_n)(x)\,.$$

Results: RG as mapping cumulants

$$p_{n+1}(x) = \sqrt{2} \int dy\, p_n(\sqrt{2}x - y)\, p_n(y) = (\mathscr{R} p_n)(x)\,.$$

Results: RG as mapping cumulants

$$p_{n+1}(x) = \sqrt{2} \int dy\, p_n(\sqrt{2}x - y)\, p_n(y) = (\mathscr{R} p_n)(x)\,.$$

Cumulant generating function: $\mathscr{W}_{n+1}(s) = \ln \int_{-\infty}^{\infty} e^{sx}\, p_{n+1}(x)\, dx\,.$

Results: RG as mapping cumulants

$$p_{n+1}(x) = \sqrt{2} \int dy\, p_n(\sqrt{2}x - y)\, p_n(y) = (\mathscr{R} p_n)(x)\,.$$

Cumulant generating function: $\mathscr{W}_{n+1}(s) = \ln \int_{-\infty}^{+\infty} e^{sx}\, p_{n+1}(x)\, dx\,.$

Results: RG as mapping cumulants

$$p_{n+1}(x) = \sqrt{2} \int dy\, p_n(\sqrt{2}x - y)\, p_n(y) = (\mathscr{R} p_n)(x)\,.$$

Cumulant generating function: $\mathscr{W}_{n+1}(s) = \ln \int_{-\infty}^{\infty} e^{sx}\, p_{n+1}(x)\, dx\,.$

Results: RG as mapping cumulants

$$p_{n+1}(x) = \sqrt{2} \int dy\, p_n(\sqrt{2}x - y)\, p_n(y) = (\mathscr{R} p_n)(x)\,.$$

Cumulant generating function: $\mathscr{W}_{n+1}(s) = \ln \int_{-\infty}^{\infty} e^{sx}\, p_{n+1}(x)\, dx\,.$

Obtaining $\kappa_r^{(n+1)} = 2^{1-r/2} \kappa_r^{(n)}$ for $r \geq 1$.

Results: Symmetry and linearity

$$p_{n+1}(x) = \sqrt{2}\int dy\, p_n(\sqrt{2}x - y)\, p_n(y) = (\mathscr{R}p_n)(x)\,.$$

Results: Symmetry and linearity

Symmetry and linearity

$$\phi(x) = x$$

Solution:

$$w_0 = \frac{1}{w_2\sqrt{2}} - w_1$$

$$b_2 = -2w_2 b_1 \,.$$

$$p_{n+1}(x) = \sqrt{2}\int dy\, p_n(\sqrt{2}x - y)\, p_n(y) = (\mathscr{R}p_n)(x)\,.$$

Results: Symmetry and linearity

Symmetry and linearity

$\phi(x) = x$

Solution:

$$w_0 = \frac{1}{w_2\sqrt{2}} - w_1$$

$$b_2 = -2w_2 b_1 \,.$$

Symmetry and (minimal) non-linearity

$\phi(x) = x + \alpha x^2$

Solution:

$$2w_2(w_0 + w_1) = \sqrt{2}$$
$$2w_2\alpha(w_0 + w_1)^2 = 0$$
$$2w_2\alpha(w_0^2 + w_1^2) = 0.$$

$$p_{n+1}(x) = \sqrt{2}\int dy\, p_n(\sqrt{2}x - y)\, p_n(y) = (\mathscr{R}p_n)(x)\,.$$

Results: Symmetry and linearity

Symmetry and linearity

$\phi(x) = x$

Solution:

$$w_0 = \frac{1}{w_2\sqrt{2}} - w_1$$

$$b_2 = -2w_2 b_1 .$$

Symmetry and (minimal) non-linearity

$\phi(x) = x + \alpha x^2$

Solution:

Inconsistent

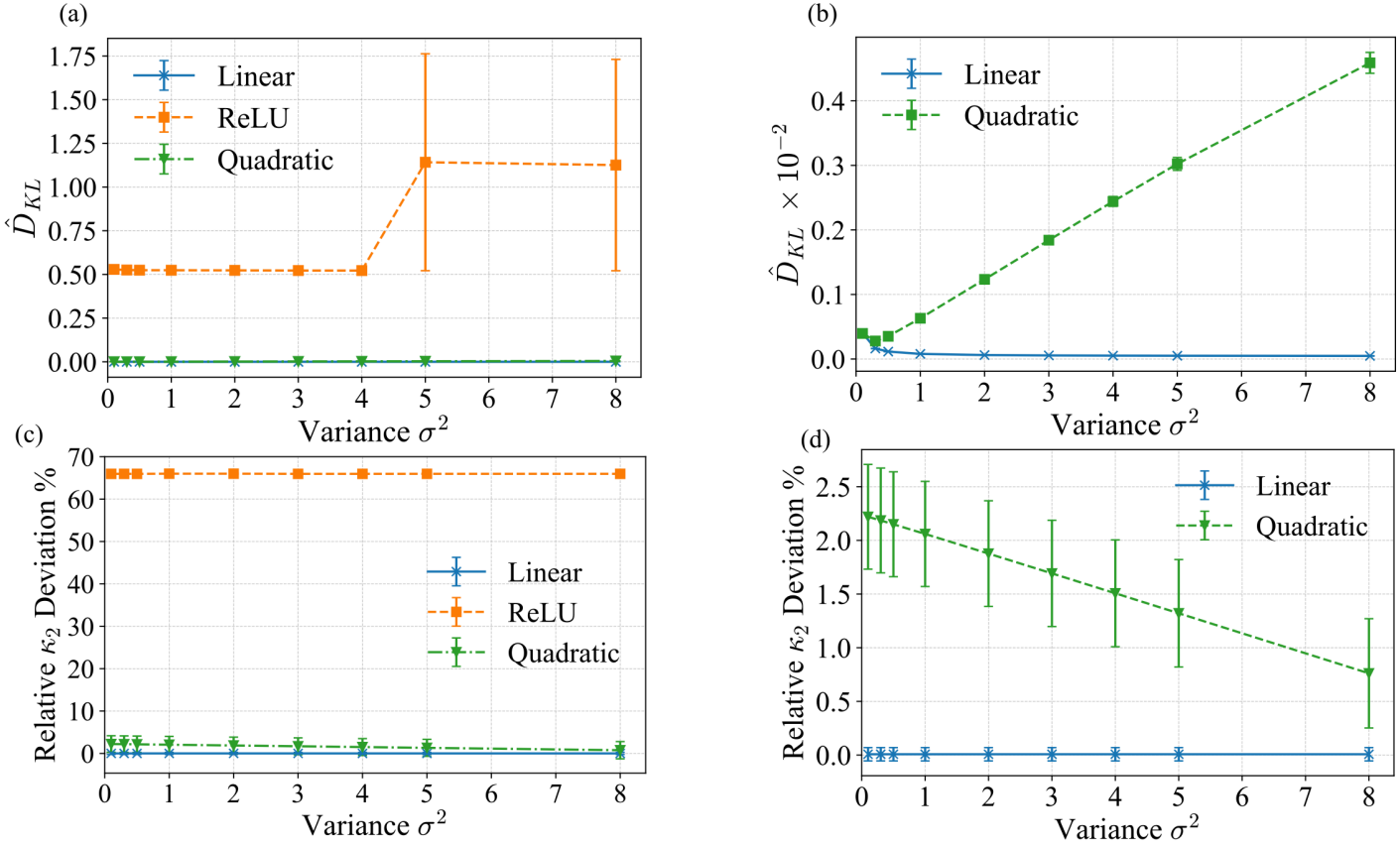
$2w_2($

$2w_2\alpha(w_0 + w_1)^2 = 0$

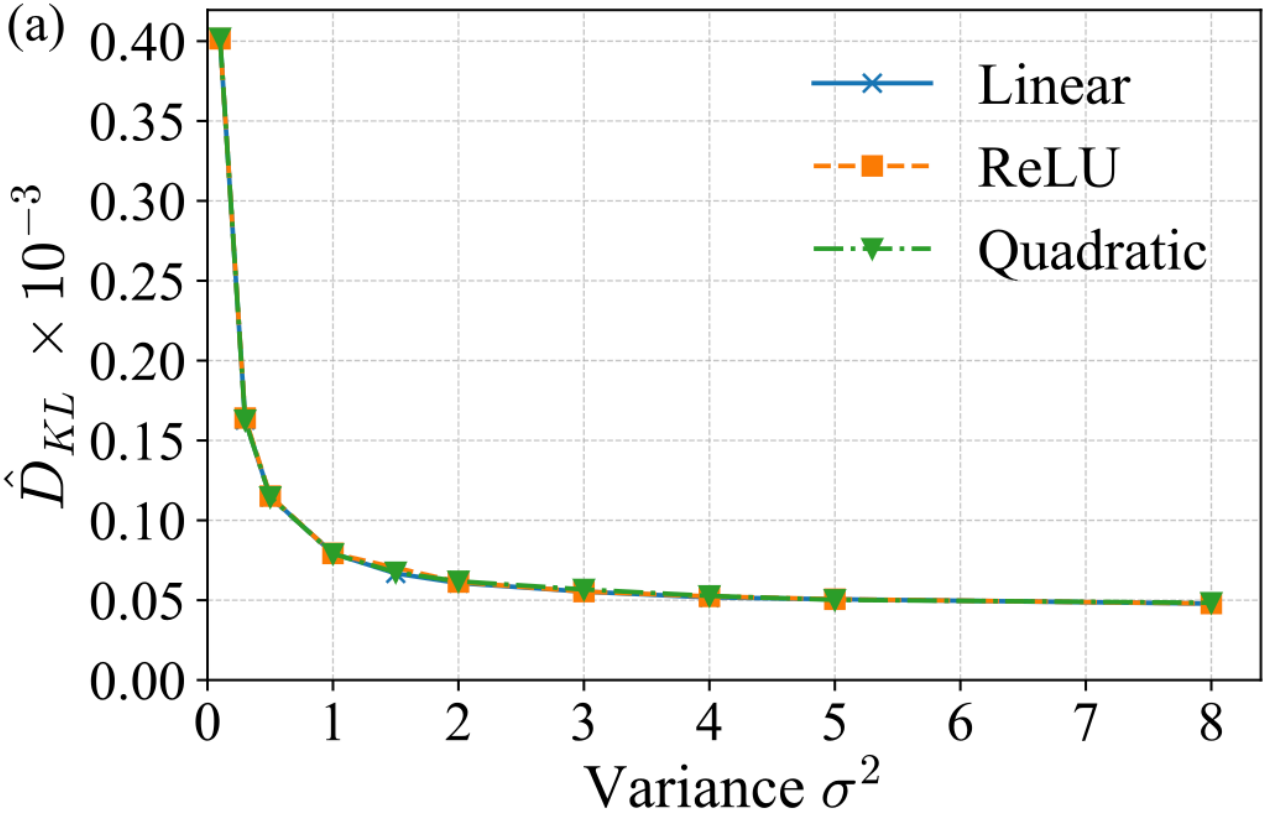
$2w_2\alpha(w_0^2 + w_1^2) = 0.$

$$p_{n+1}(x) = \sqrt{2}\int dy\, p_n(\sqrt{2}x - y)\, p_n(y) = (\mathscr{R}p_n)(x) .$$

Results: Only Symmetry



Results: No constraints



Limitations and Future Directions

- Limitations
- CLT is a solvable **toy RG flow**.
- We didn’t use **encoders/decoders**.

- Future Work
- We are testing on **more complex architectures**.
- We are analysing the **Hubbard model**.

Conclusions

- Are NNs representing and RG flow? **Yes (in Jona-Lasinio formalism)**

- What can we learn from the RG flow framework? **To not focus only on symmetry but also on the order of mixing required**

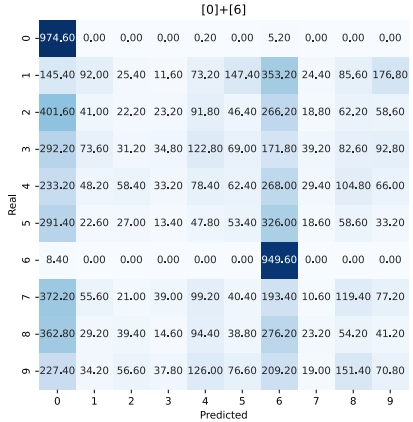
PAPER

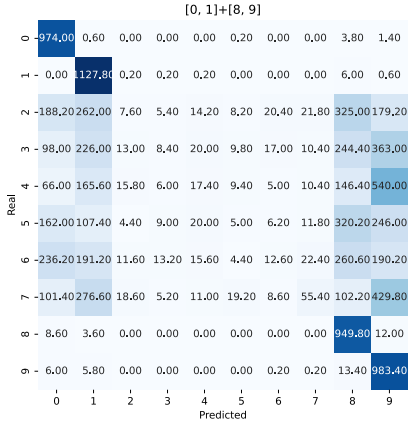
Submodules via Entropy

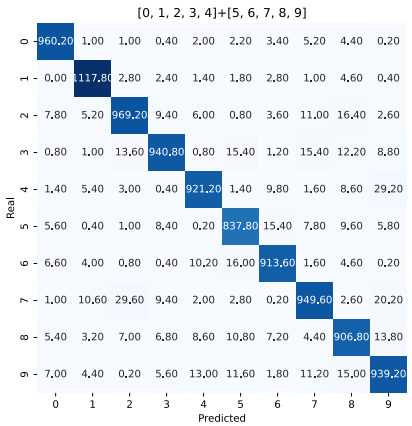
Composable Sparse Subnetworks via Maximum-Entropy Principle

F. Caso, S. Fonio, N. Saccomanno, S. Monaco, F. Silvestri

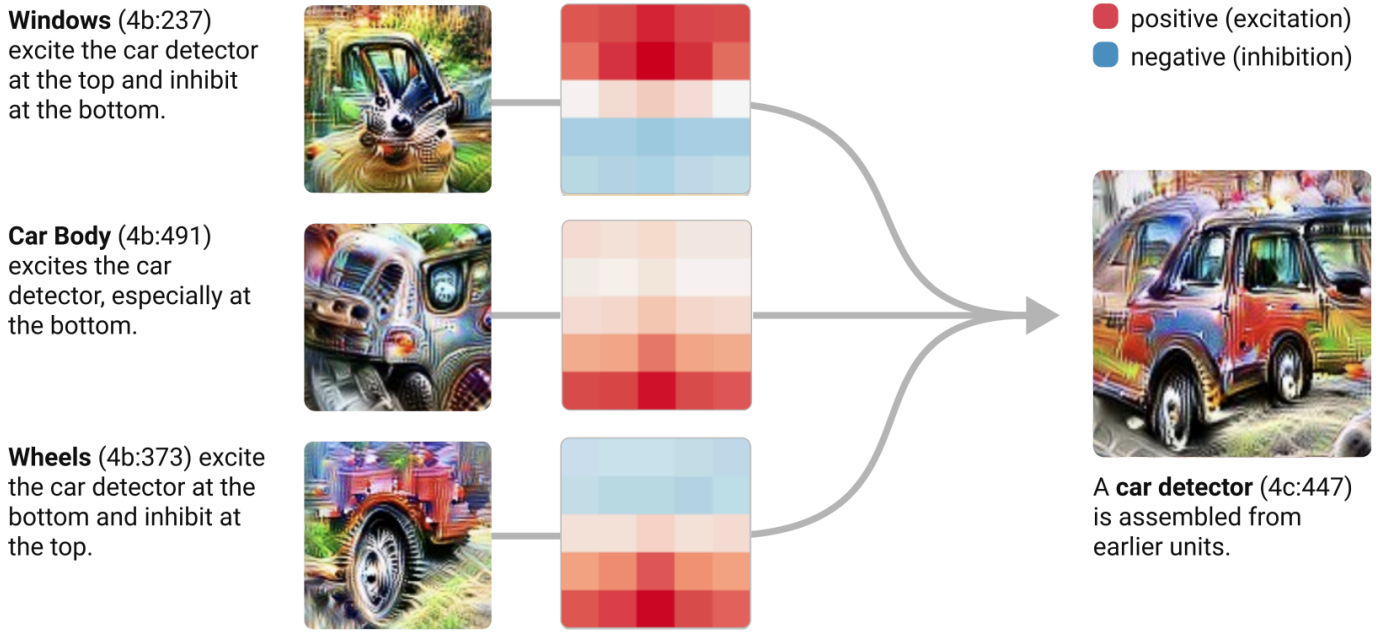
NeurIPS 2025 Workshop on Mechanistic Interpretability







Motivation: Circuits



Olah et al.

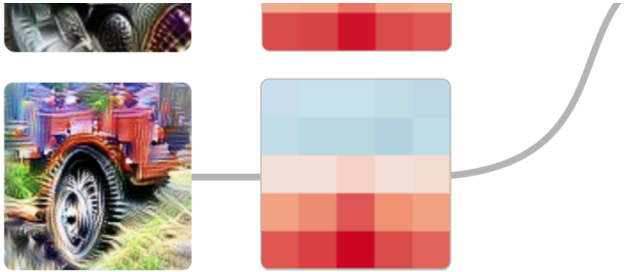
Motivation: Circuits

Windows (4b:237) excite the car detector at the top and inhibit at the bottom.

positive (excitation)
negative (inhibition)

Spurious circuits entangled to other classes

Wheels (4b:373) excite the car detector at the bottom and inhibit at the top.



A **car detector** (4c:447) is assembled from earlier units.

Olah et al.

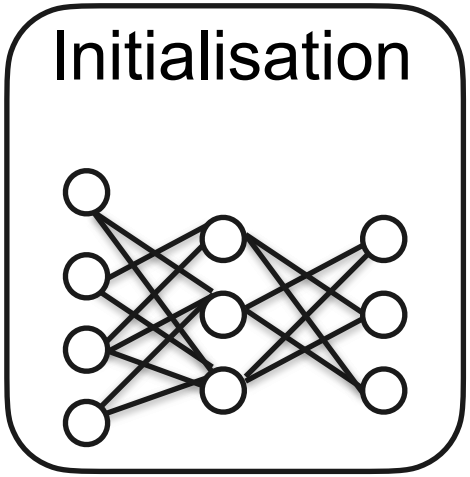
Can we train class-specialised subnetworks that remain ignorant outside their domain and compose into generalist model?

The lottery ticket hypothesis

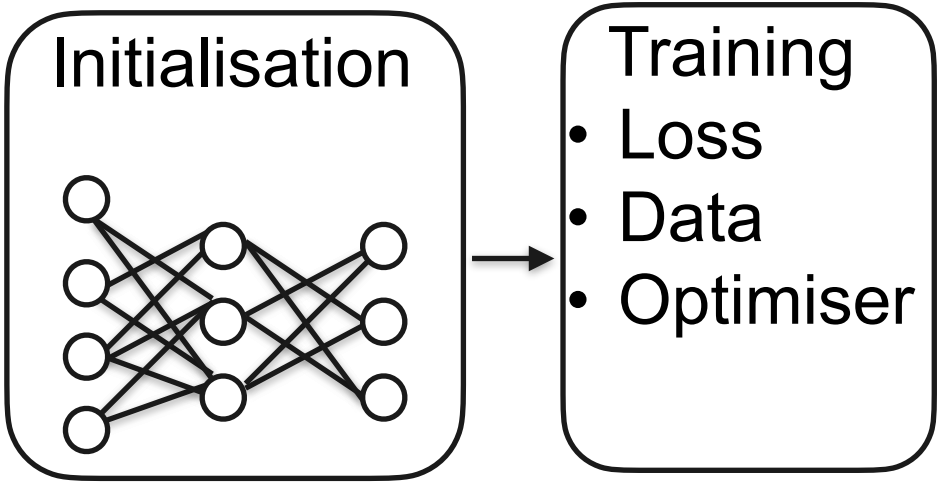
Frankle and Carbin proposed that

A randomly-initialized, dense neural network contains a subnetwork that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations.

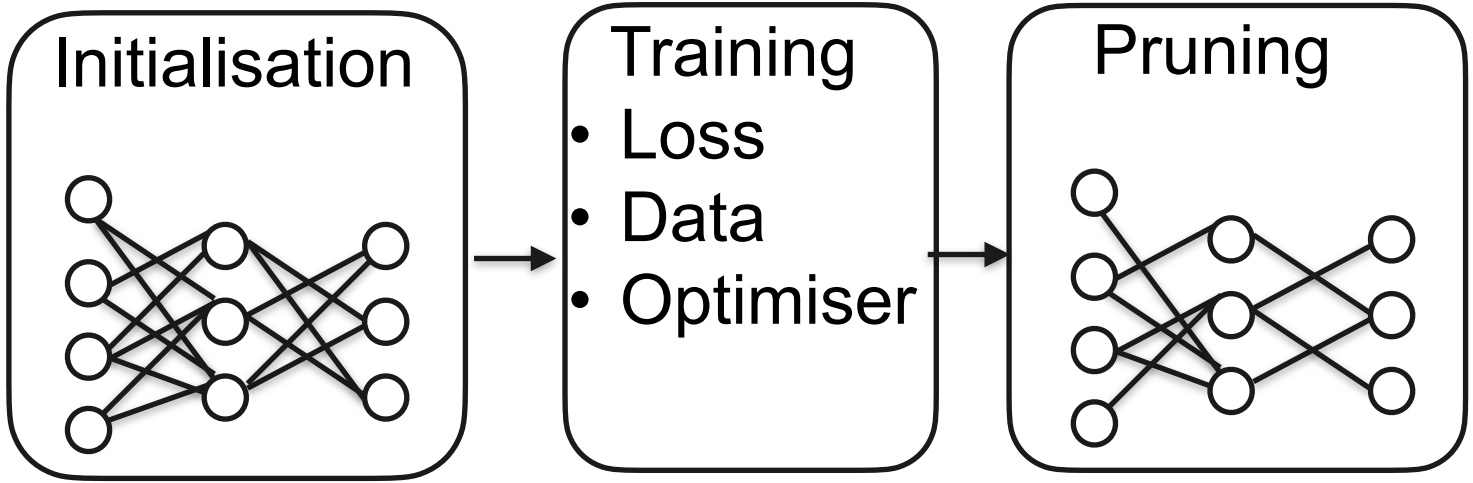
The lottery ticket hypothesis



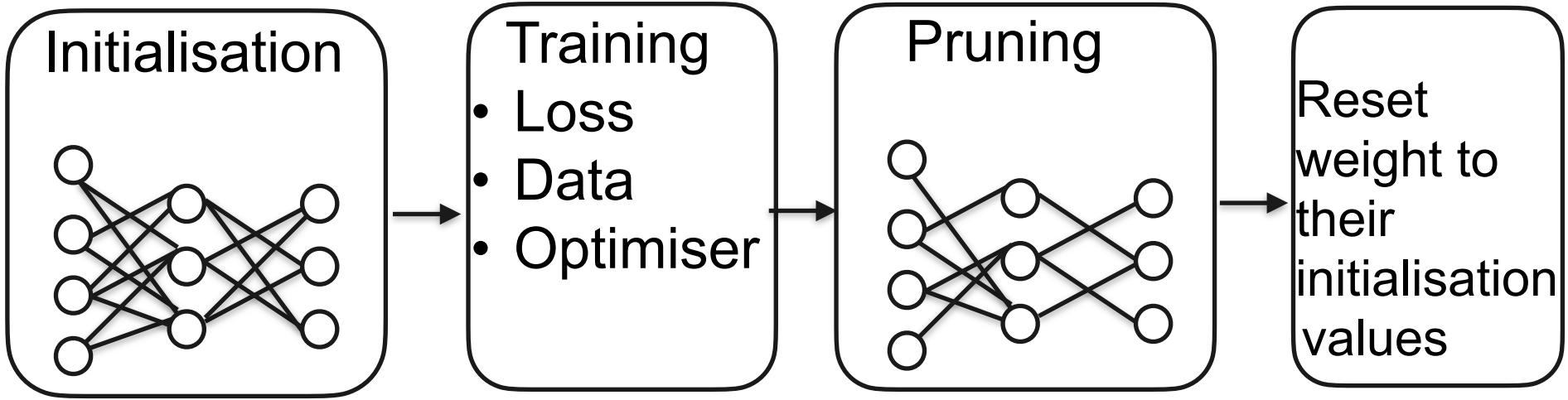
The lottery ticket hypothesis



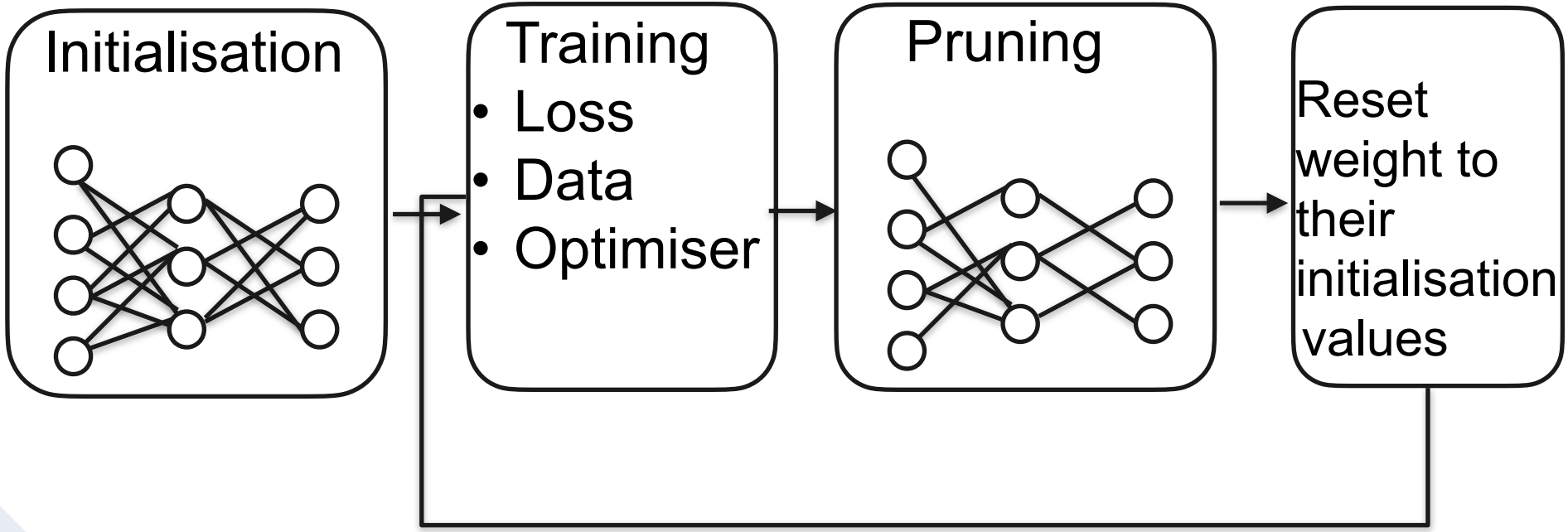
The lottery ticket hypothesis



The lottery ticket hypothesis



The lottery ticket hypothesis



Maximum Entropy Principle

Information Theory and Statistical Mechanics

E. T. JAYNES
Department of Physics, Stanford University, Stanford, California
(Received September 4, 1956; revised manuscript received March 4, 1957)

Maximum Entropy Principle

Between the distributions that satisfy given constraints the most agnostic one is the one that maximazies entropy.

Information Theory and Statistical Mechanics

E. T. JAYNES
Department of Physics, Stanford University, Stanford, California
(Received September 4, 1956; revised manuscript received March 4, 1957)

Our framework: The MaxEnt Loss

Let C be the full set of classes and $R \subseteq C$ the set of *rewarded classes*

Our framework: The MaxEnt Loss

Let C be the full set of classes and $R \subseteq C$ the set of *rewarded classes*

For a training sample (x, y), where $y \in C$,

Our framework: The MaxEnt Loss

Let C be the full set of classes and $R \subseteq C$ the set of *rewarded classes*

For a training sample (x, y), where $y \in C$, we define the

target distribution $\tilde{y} \in \mathbb{R}^{|c|}$ *as*

$$\tilde{y}_i = \begin{cases} \delta_{i=y} & \text{if } y \in R \\ \frac{1}{|C|} & \text{otherwise} \end{cases}$$

Our framework: The MaxEnt Loss

$$\tilde{y}_i = \begin{cases} \delta_{i=y} & \text{if } y \in R \\ \dfrac{1}{|C|} & \text{otherwise} \end{cases}$$

If $C = \{0,1,2\}$ and $R = \{0\}$

Our framework: The MaxEnt Loss

$$\tilde{y}_i = \begin{cases} \delta_{i=y} & \text{if } y \in R \\ \dfrac{1}{|C|} & \text{otherwise} \end{cases}$$

If $C = \{0,1,2\}$ and $R = \{0\}$

$\tilde{y} = (1,0,0)$ for class 0 and

Our framework: The MaxEnt Loss

$$\tilde{y}_i = \begin{cases} \delta_{i=y} & \text{if } y \in R \\ \dfrac{1}{|C|} & \text{otherwise} \end{cases}$$

If $C = \{0,1,2\}$ and $R = \{0\}$

$\tilde{y} = (1,0,0)$ for class 0 and

$\tilde{y} = (0.33,0.33,0.33)$ for classes 1 and 2

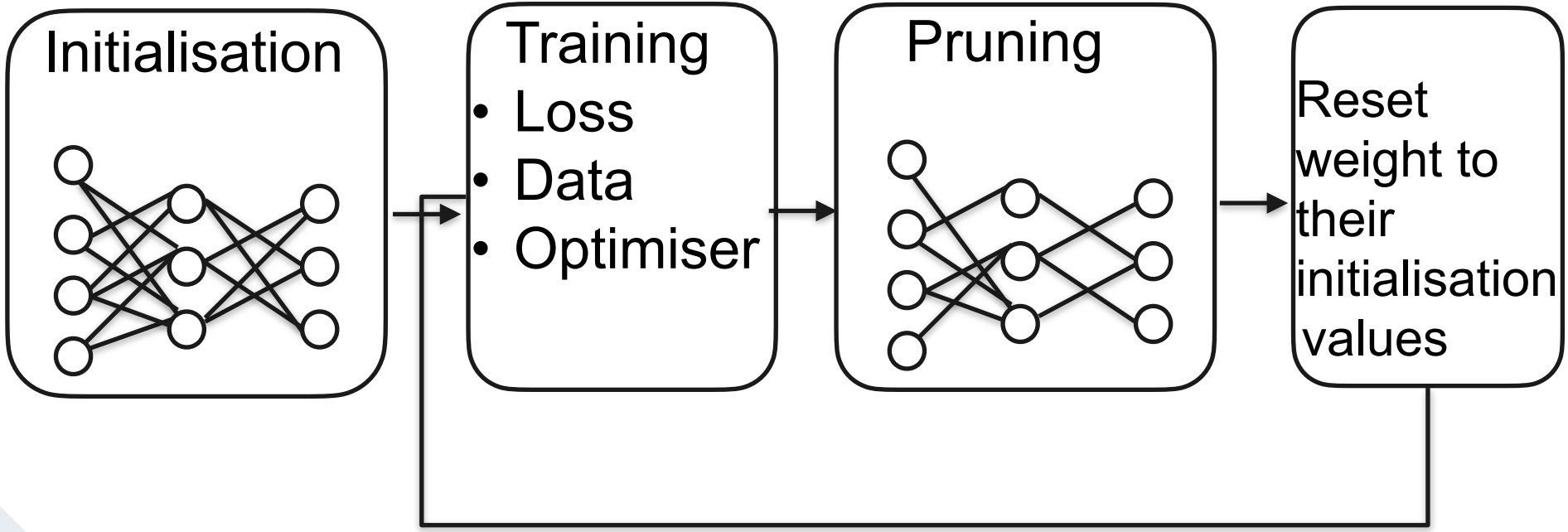
Our framework: The MaxEnt Loss

$$\tilde{y}_i = \begin{cases} \delta_{i=y} & \text{if } y \in R \\ \dfrac{1}{|C|} & \text{otherwise} \end{cases}$$

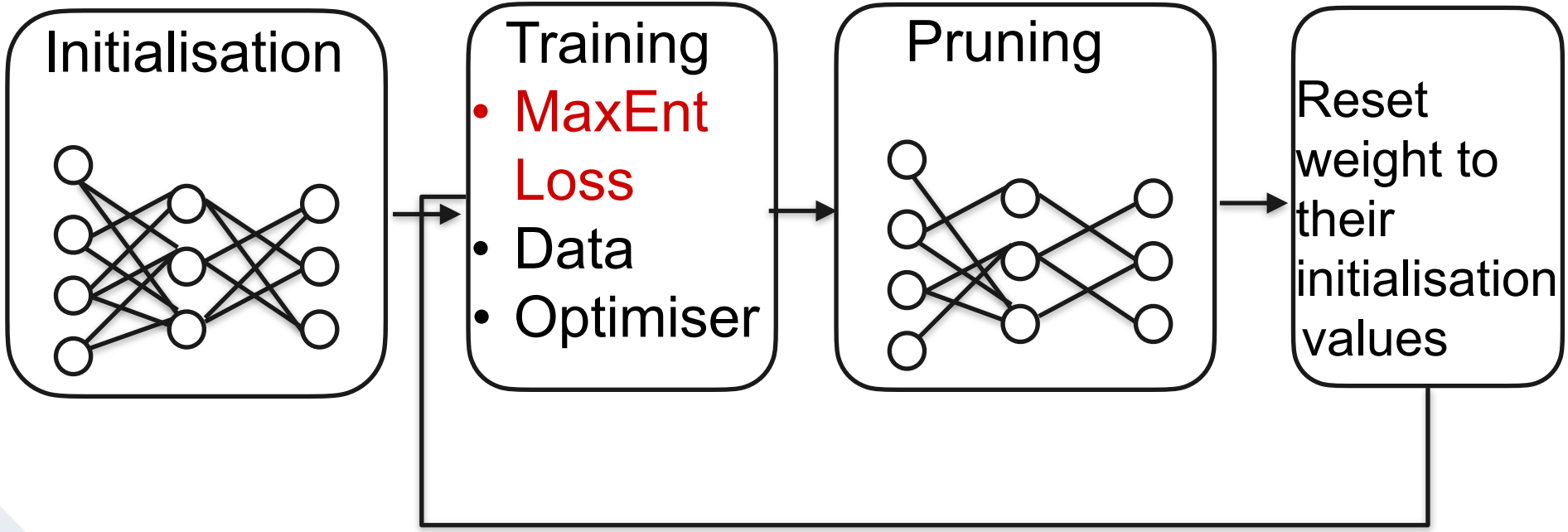
$$\mathscr{L}_{\text{ME}}(x, y) = \text{KL}(\tilde{y} \parallel \hat{y}) = \sum_{i=1}^{|\mathscr{C}|} \tilde{y}_i \log\left(\frac{\tilde{y}_i}{\hat{y}_i}\right)$$

With $\hat{y} = \text{softmax}(f_\theta(x))$

Original IMP

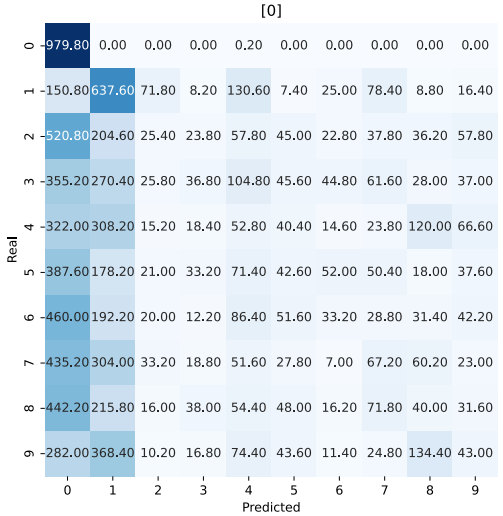


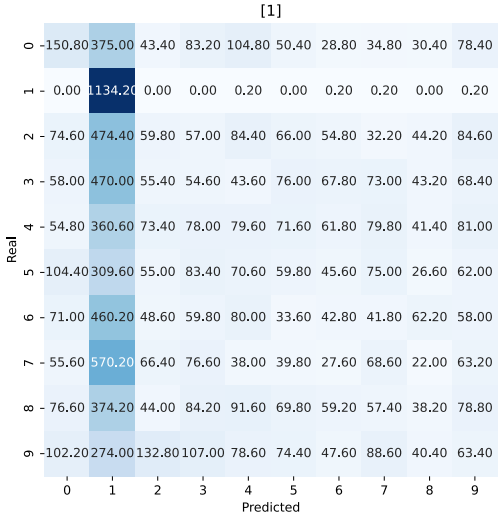
Our framework: The MaxEnt Loss

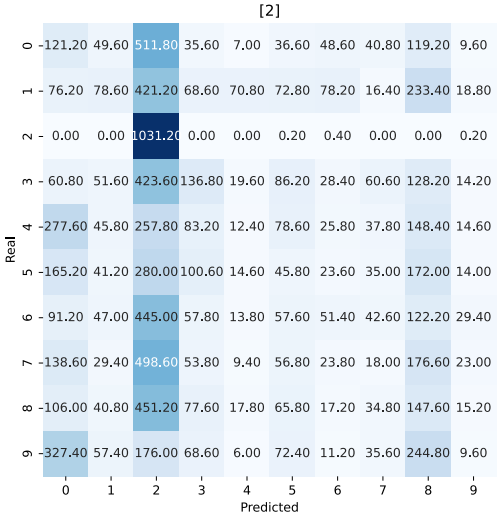


Sanity check: do the modules specialise?

Results: specialization







Results: specialization

Table 1: Single submodule behaviour when using MaxEnt Loss with and without IMP.

Model	IMP	MNIST		Fashion MNIST		HAR		Yeast	
		Entropy	Rewarded Acc	Entropy	Rewarded Acc	Entropy	Rewarded Acc	Entropy	Rewarded Acc
Shallow MLP	No	2.296 (0.003)	0.998 (0.002)	2.296 (0.003)	0.998 (0.002)	1.762 (0.017)	0.997 (0.007)	1.298 (0.062)	0.995 (0.009)
	Yes	2.293 (0.004)	0.999 (0.001)	2.293 (0.004)	0.999 (0.001)	1.757 (0.023)	0.996 (0.008)	1.297 (0.059)	0.998 (0.006)
Deep MLP	No	2.298 (0.002)	0.997 (0.003)	2.285 (0.013)	0.995 (0.004)	1.772 (0.014)	0.992 (0.013)	1.302 (0.064)	0.996 (0.009)
	Yes	2.300 (0.001)	0.998 (0.002)	2.291 (0.008)	0.991 (0.007)	1.762 (0.023)	0.999 (0.005)	1.302 (0.056)	1.000 (0.000)
CNN	No	2.302 (0.000)	0.998 (0.004)	2.302 (0.000)	0.996 (0.004)	-	-	-	-
	Yes	2.302 (0.000)	0.994 (0.005)	2.302 (0.000)	0.992 (0.005)	-	-	-	-

Can submodules be composed?

Mode connectivity

Following Frankle et al. and Lubana et al., we say that θ_1 and θ_2 are *mode connected* along a path $\gamma(t)$ if:

Mode connectivity

Following Frankle et al. and Lubana et al., we say that θ_1 and θ_2 are *mode connected* along a path $\gamma(t)$ if:

$$\forall t \in [0,1], \quad \mathscr{L}(f_{\gamma(t)}(\mathscr{D})) \leq (1-t)\mathscr{L}(f_{\theta_1}(\mathscr{D})) + t\mathscr{L}(f_{\theta_2}(\mathscr{D})) + \epsilon$$

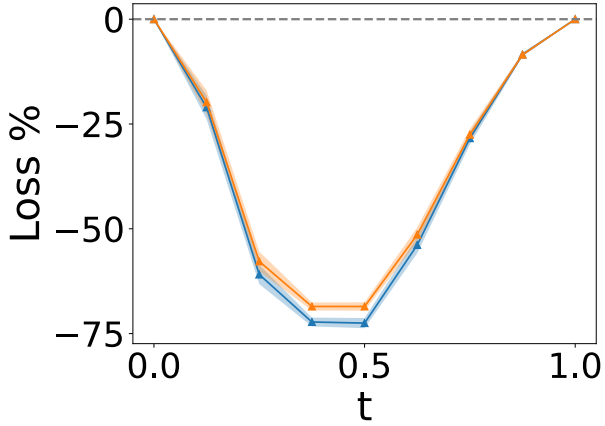
Mode connectivity

Following Frankle et al. and Lubana et al., we say that θ_1 and θ_2 are *mode connected* along a path $\gamma(t)$ if:

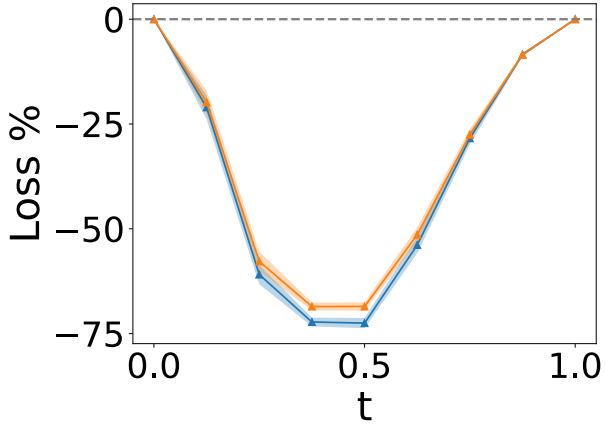
$$\forall t \in [0,1], \quad \mathscr{L}(f_{\gamma(t)}(\mathscr{D})) \leq (1-t)\mathscr{L}(f_{\theta_1}(\mathscr{D})) + t\mathscr{L}(f_{\theta_2}(\mathscr{D})) + \epsilon$$

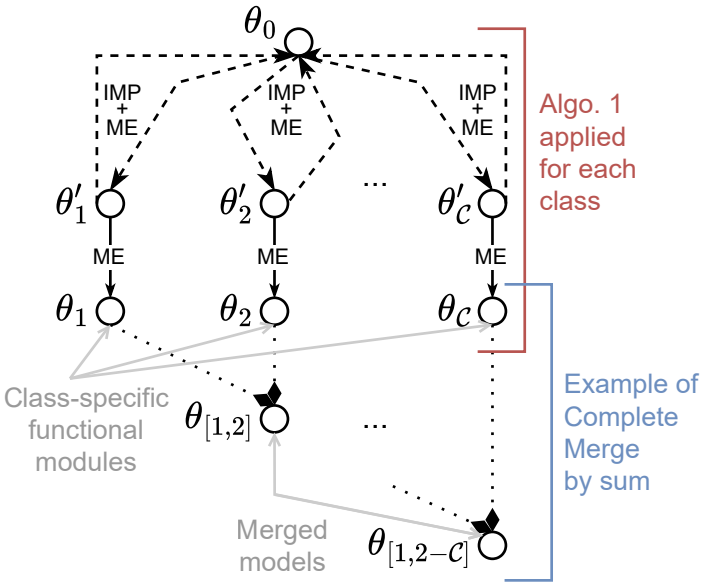
$$\gamma_{\theta_1 \to \theta_2}(t) = \begin{cases} \theta_1 + 2t \cdot \theta_2 & \text{if } t \leq 0.5 \\ 2(1-t) \cdot \theta_1 + \theta_2 & \text{if } t > 0.5 \end{cases}$$

Our framework: Model merging

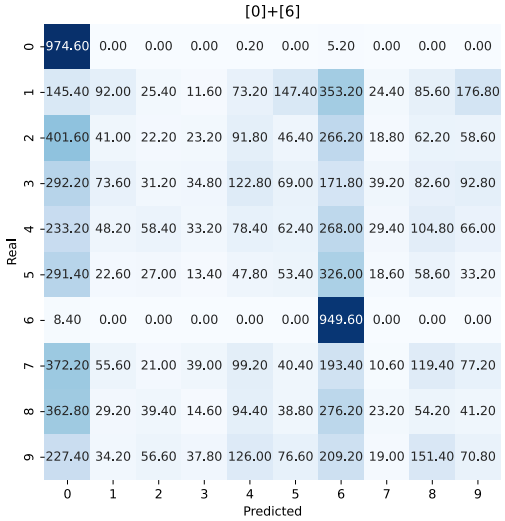


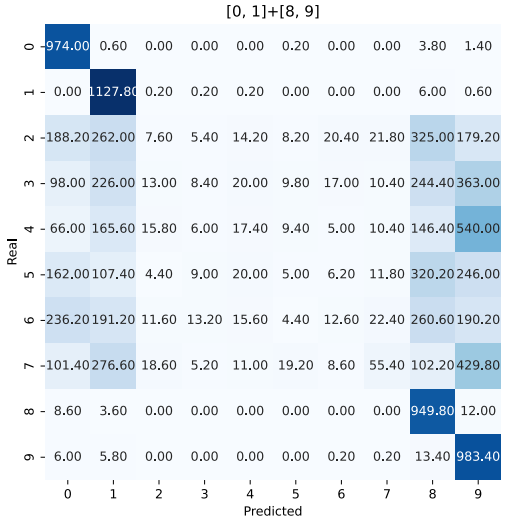
Our framework: Model merging

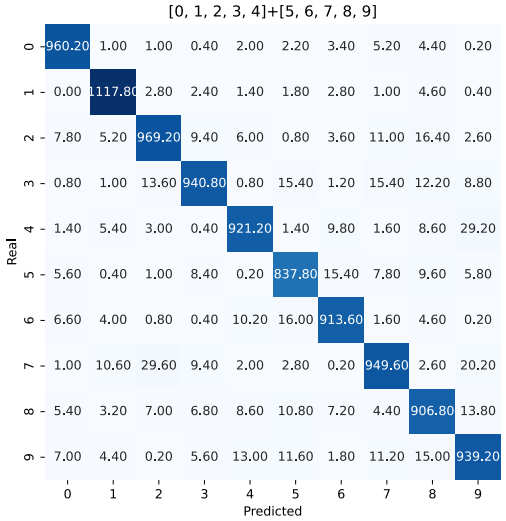




Our framework: Model merging

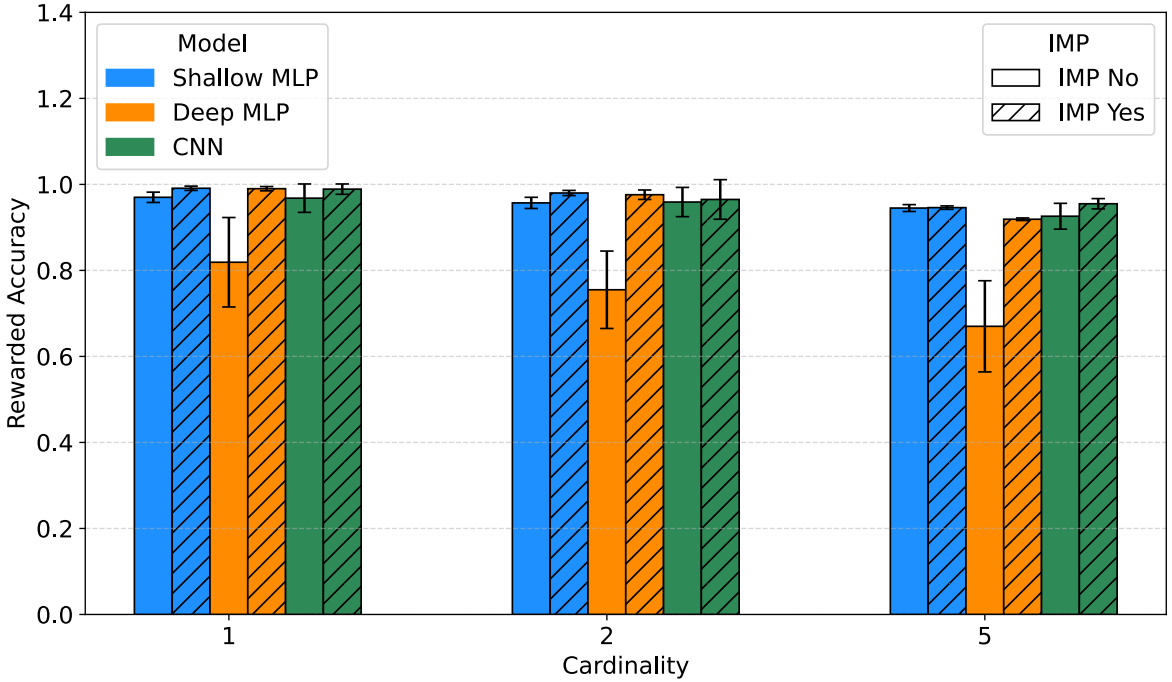






What’s the effect of the IMP procedure?

Results: IMP procedure



Can we quantify the effect of the MaxEnt loss?

Our framework: baselines

CrossEntropy:

In this case the model is exposed only to R and not to the whole C.

Our framework: baselines

CrossEntropy:

In this case the model is exposed only to R and not to the whole C.

Quasi-MaxEnt:

$$\tilde{y}_i = \begin{cases} \delta_{i=y} & \text{if } y \in \mathscr{R} \\ \delta_{i \neq j, j \in \mathscr{R}} \frac{1}{|\mathscr{C} \setminus \mathscr{R}|} & \text{otherwise} \end{cases}$$

Results: MaxEnt Loss

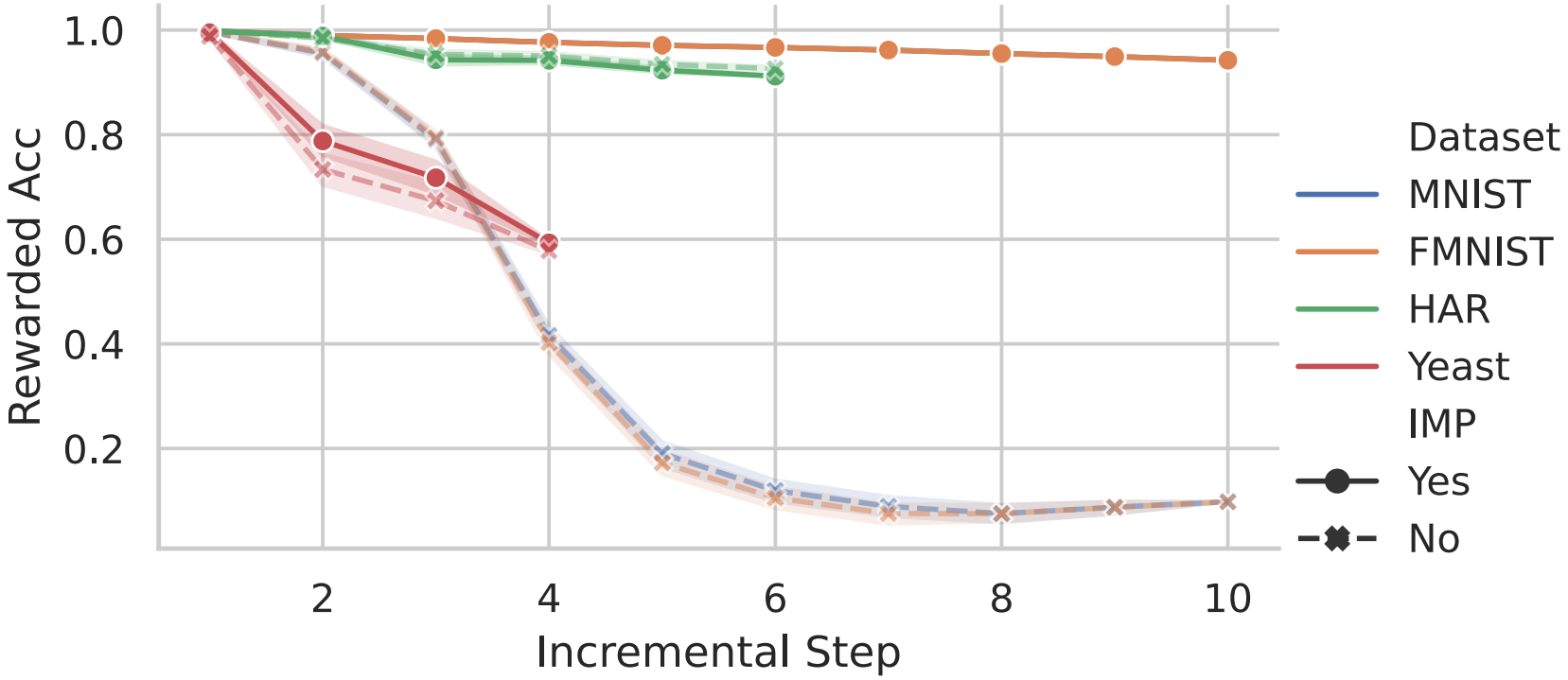
Model	$\lvert\mathcal{R}\rvert$	Loss	FMNIST Entropy	FMNIST R-Acc	MNIST Entropy	MNIST R-Acc	HAR Entropy	HAR R-Acc	Yeast Entropy	Yeast R-Acc
MLP	1	XE	0.183 (0.151)	0.877 (0.186)	0.507 (0.150)	0.708 (0.179)	1.255 (0.191)	0.855 (0.197)	1.378 (0.004)	0.631 (0.101)
		QME	2.031 (0.132)	0.908 (0.105)	2.082 (0.005)	0.973 (0.014)	1.196 (0.073)	0.946 (0.046)	1.056 (0.082)	0.799 (0.106)
		ME	2.287 (0.005)	**0.977** (0.005)	2.287 (0.005)	**0.991** (0.005)	1.716 (0.033)	**0.985** (0.016)	1.146 (0.077)	**0.853** (0.127)
	2	XE	0.277 (0.108)	0.786 (0.141)	0.332 (0.080)	0.838 (0.074)	1.167 (0.217)	0.871 (0.055)	–	0.381 (0.036)
		QME	1.824 (0.207)	0.917 (0.049)	1.683 (0.216)	0.959 (0.016)	1.013 (0.223)	0.891 (0.024)	–	0.559 (0.011)
		ME	2.267 (0.007)	**0.977** (0.005)	2.268 (0.007)	**0.980** (0.006)	1.670 (0.045)	**0.935** (0.027)	–	**0.616** (0.010)
	5	XE	–	0.480 (0.075)	–	0.842 (0.027)	–	–	–	–
		QME	–	0.741 (0.026)	–	0.905 (0.023)	–	–	–	–
		ME	–	**0.946** (0.004)	–	**0.946** (0.004)	–	–	–	–
Deep MLP	1	XE	0.215 (0.150)	0.802 (0.205)	0.422 (0.190)	0.689 (0.194)	1.059 (0.313)	0.840 (0.210)	1.386 (0.0003)	0.553 (0.144)
		QME	1.842 (0.329)	0.868 (0.176)	2.061 (0.046)	0.954 (0.045)	1.098 (0.128)	0.819 (0.155)	0.883 (0.118)	0.783 (0.114)
		ME	2.245 (0.072)	**0.975** (0.039)	2.291 (0.031)	**0.990** (0.005)	1.697 (0.050)	**0.988** (0.016)	1.063 (0.109)	**0.859** (0.120)
	2	XE	0.166 (0.083)	0.678 (0.081)	0.208 (0.054)	0.743 (0.111)	1.065 (0.344)	0.824 (0.068)	–	0.382 (0.046)
		QME	1.347 (0.484)	0.863 (0.088)	1.224 (0.352)	0.912 (0.038)	0.665 (0.251)	0.787 (0.093)	–	0.562 (0.023)
		ME	2.197 (0.095)	**0.898** (0.045)	2.275 (0.026)	**0.976** (0.011)	1.607 (0.134)	**0.930** (0.028)	–	**0.589** (0.018)
CNN	1	XE	0.016 (0.026)	0.557 (0.150)	0.041 (0.128)	0.529 (0.080)	–	–	–	–
		QME	0.981 (0.322)	0.880 (0.136)	1.001 (0.377)	0.961 (0.078)	–	–	–	–
		ME	2.297 (0.039)	**0.960** (0.097)	2.286 (0.057)	**0.989** (0.012)	–	–	–	–
	2	XE	0.052 (0.038)	0.604 (0.169)	0.070 (0.042)	0.624 (0.138)	–	–	–	–
		QME	0.705 (0.397)	0.888 (0.071)	0.312 (0.307)	0.928 (0.074)	–	–	–	–
		ME	1.984 (0.304)	**0.930** (0.044)	1.950 (0.456)	**0.965** (0.046)	–	–	–	–

Results: MaxEnt Loss

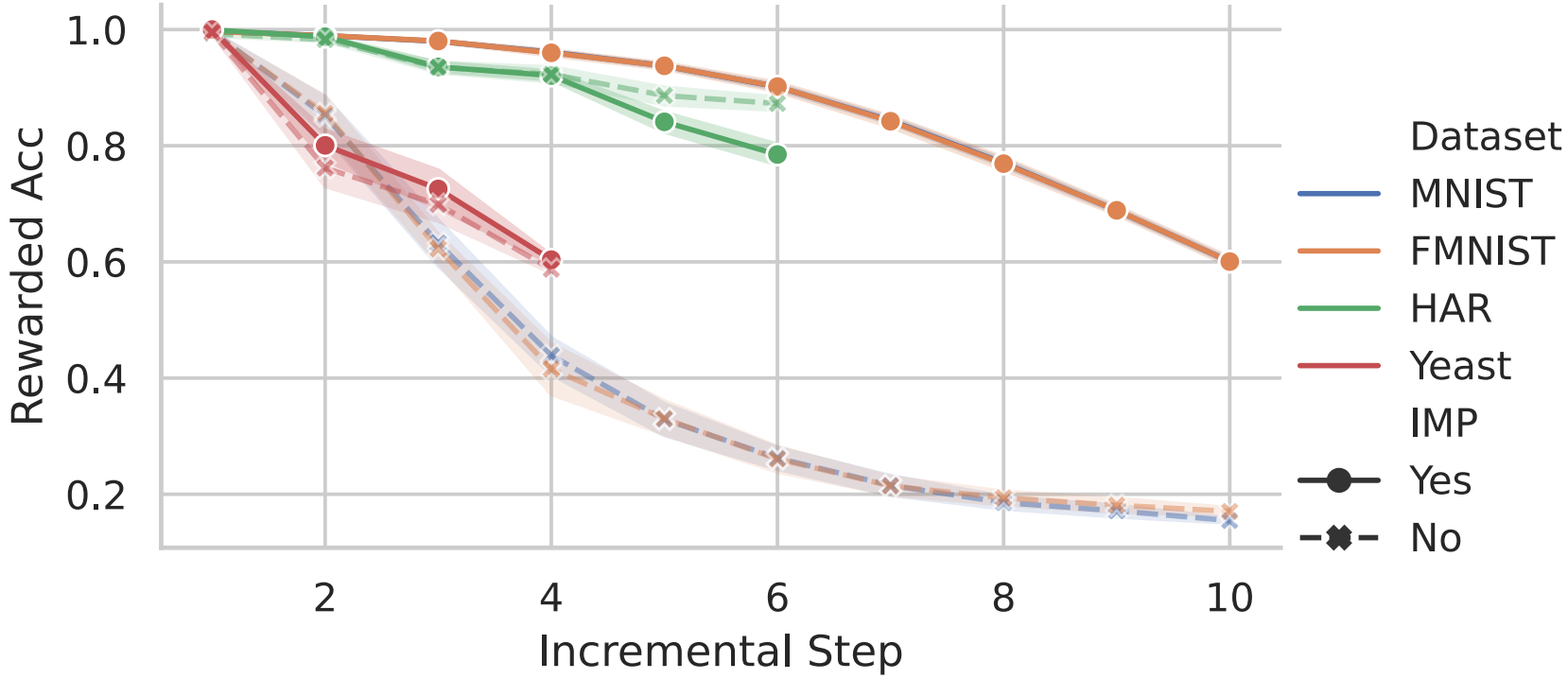
Model	$\lvert\mathcal{R}\rvert$	Loss	FMNIST Entropy	FMNIST R-Acc	MNIST Entropy	MNIST R-Acc	HAR Entropy	HAR R-Acc	Yeast Entropy	Yeast R-Acc
MLP	1	XE	0.183 (0.151)	0.877 (0.186)	0.507 (0.150)	0.708 (0.179)	1.255 (0.191)	0.855 (0.197)	1.378 (0.004)	0.631 (0.101)
		QME	2.031 (0.132)	0.908 (0.105)	2.082 (0.005)	0.973 (0.014)	1.196 (0.073)	0.946 (0.046)	1.056 (0.082)	0.799 (0.106)
		ME	2.287 (0.005)	**0.977** (0.005)	2.287 (0.005)	**0.991** (0.005)	1.716 (0.033)	**0.985** (0.016)	1.146 (0.077)	**0.853** (0.127)
	2	XE	0.277 (0.108)	0.786 (0.141)	0.332 (0.080)	0.838 (0.074)	1.167 (0.217)	0.871 (0.055)	–	0.381 (0.036)
		QME	1.824 (0.207)	0.917 (0.049)	1.683 (0.216)	0.959 (0.016)	1.013 (0.223)	0.891 (0.024)	–	0.559 (0.011)
		ME	2.267 (0.007)	**0.977** (0.005)	2.268 (0.007)	**0.980** (0.006)	1.670 (0.045)	**0.935** (0.027)	–	**0.616** (0.010)
	5	XE	–	0.480 (0.075)	–	0.842 (0.027)	–	–	–	–
		QME	–	0.741 (0.026)	–	0.905 (0.023)	–	–	–	–
		ME	–	**0.946** (0.004)	–	**0.946** (0.004)	–	–	–	–
Deep MLP	1	XE	0.215 (0.150)	0.802 (0.205)	0.422 (0.190)	0.689 (0.194)	1.059 (0.313)	0.840 (0.210)	1.386 (0.0003)	0.553 (0.144)
		QME	1.842 (0.329)	0.868 (0.176)	2.061 (0.046)	0.954 (0.045)	1.098 (0.128)	0.819 (0.155)	0.883 (0.118)	0.783 (0.114)
		ME	2.245 (0.072)	**0.975** (0.039)	2.291 (0.031)	**0.990** (0.005)	1.697 (0.050)	**0.988** (0.016)	1.063 (0.109)	**0.859** (0.120)
	2	XE	0.166 (0.083)	0.678 (0.081)	0.208 (0.054)	0.743 (0.111)	1.065 (0.344)	0.824 (0.068)	–	0.382 (0.046)
		QME	1.347 (0.484)	0.863 (0.088)	1.224 (0.352)	0.912 (0.038)	0.665 (0.251)	0.787 (0.093)	–	0.562 (0.023)
		ME	2.197 (0.095)	**0.898** (0.045)	2.275 (0.026)	**0.976** (0.011)	1.607 (0.134)	**0.930** (0.028)	–	**0.589** (0.018)
CNN	1	XE	0.016 (0.026)	0.557 (0.150)	0.041 (0.128)	0.529 (0.080)	–	–	–	–
		QME	0.981 (0.322)	0.880 (0.136)	1.001 (0.377)	0.961 (0.078)	–	–	–	–
		ME	2.297 (0.039)	**0.960** (0.097)	2.286 (0.057)	**0.989** (0.012)	–	–	–	–
	2	XE	0.052 (0.038)	0.604 (0.169)	0.070 (0.042)	0.624 (0.138)	–	–	–	–
		QME	0.705 (0.397)	0.888 (0.071)	0.312 (0.307)	0.928 (0.074)	–	–	–	–
		ME	1.984 (0.304)	**0.930** (0.044)	1.950 (0.456)	**0.965** (0.046)	–	–	–	–

How far can we go with the naive merge?

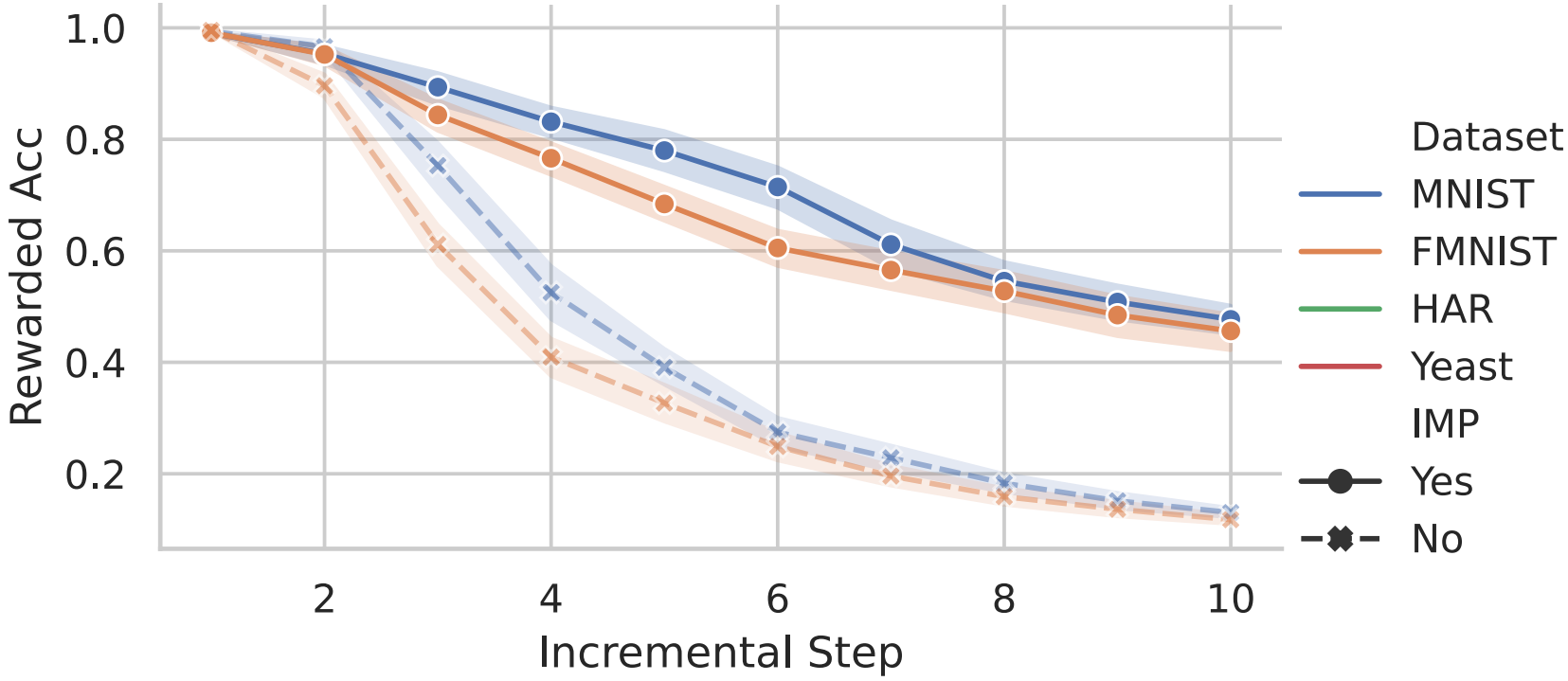
Results: Full merge on shallow MLP



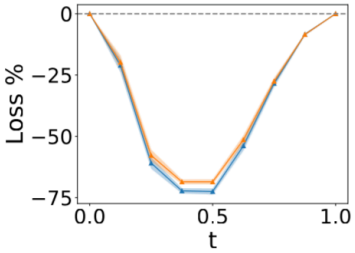
Results: Full merge on deep MLP



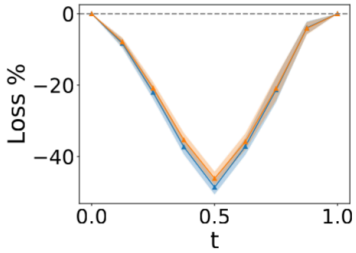
Results: Full merge on CNN



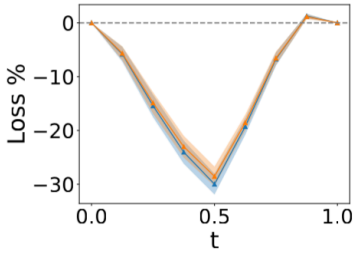
Results: Full merge



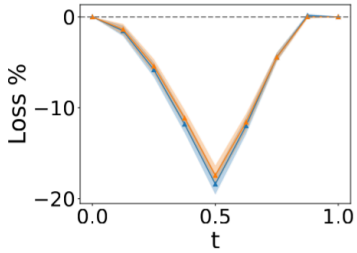
$\theta_1 = [0]$
$\theta_2 = [1]$



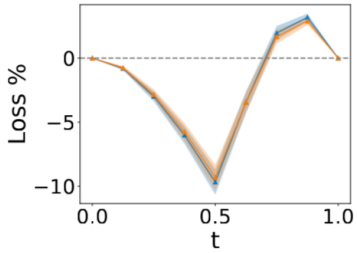
$\theta_1 = [0]+[1]+[2]$
$\theta_2 = [3]$



$\theta_1 = [0]+\ldots+[4]$
$\theta_2 = [5]$



$\theta_1 = [0]+\ldots+[6]$
$\theta_2 = [7]$



$\theta_1 = [0]+\ldots+[8]$
$\theta_2 = [9]$

Limitations and Future Directions

- Limitations
- Simple **datasets**.
- Simple **architectures**.
- Simple **merging procedure**.
- How are **submodules** and **circuits** connected?

- Future Work
- We are testing more complex **datasets**.
- We are testing more complex **architectures**.
- We are testing SOTA **merging procedures**.

Conclusions

- Can we train class-specialised subnetwork? **Yes, through the MaxEnt principle**

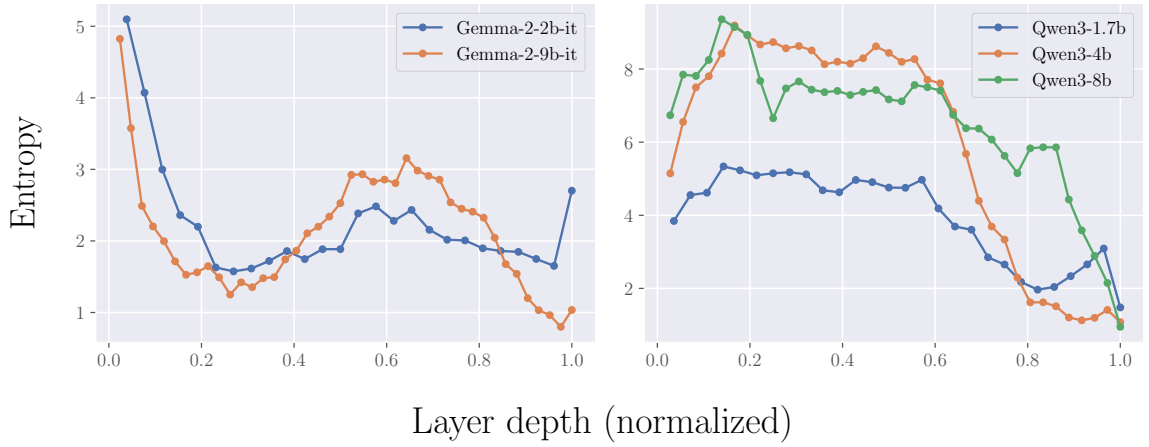
- Can we compose them via naive sum? **Yes for couple merging, more merges probably require more complex procedures**

Entropy-Lens

The Information Signature of Transformer Computations

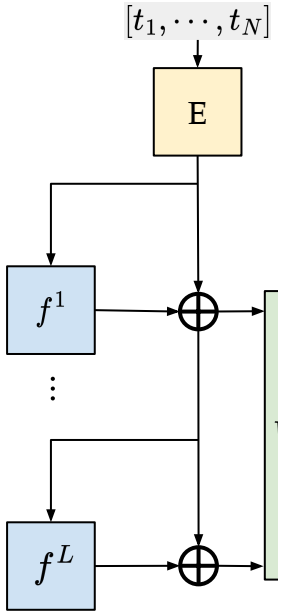
F. Caso*, R. Ali*, C. Irwin*, P. Liò

Under Review, Preprint on arXiv, * equal contribution



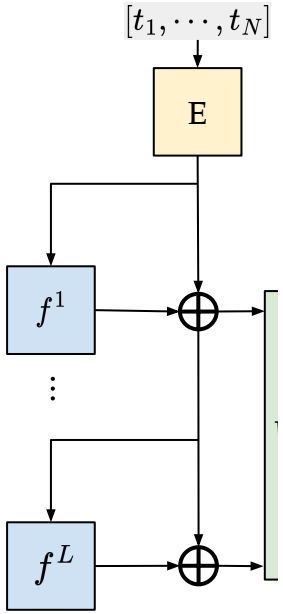
Motivation: Transformers from afar

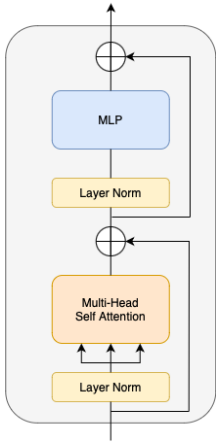
- Look from afar
  transformers are easy.



Motivation: Transformers from afar

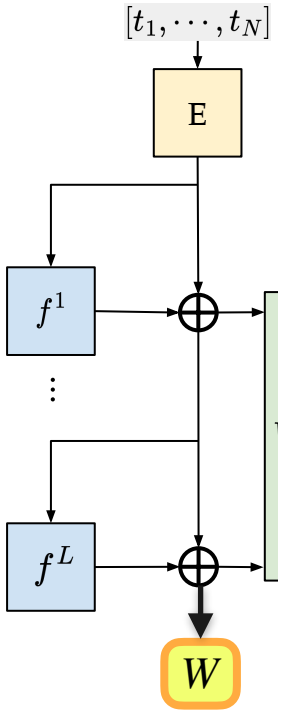
- Look from afar transformers are easy.
- Each block is composed by other elements but we don’t need to go that much into the details.





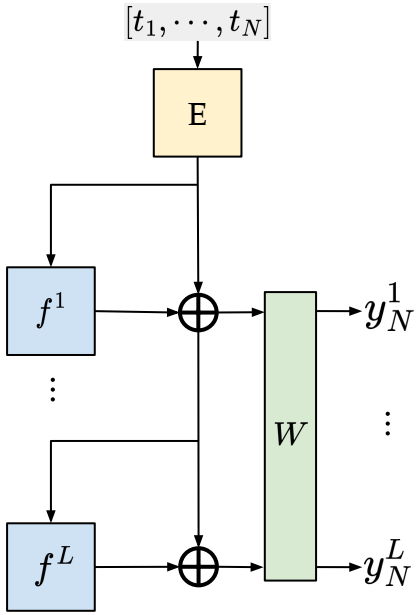
Motivation: Transformers from afar

- Look from afar
  transformers are easy.
- $W := softmax \circ D$



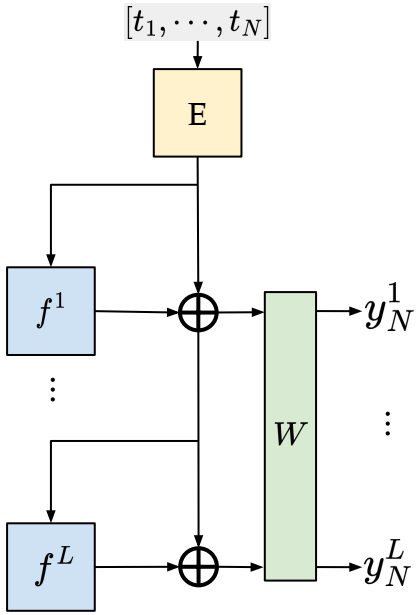
Motivation: Early-exit

- It can be helpful, depending on the task, to exit earlier



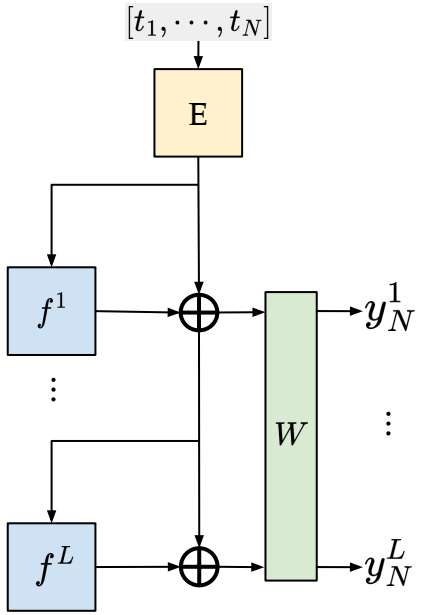
Motivation: Early-exit

- It can be helpful, depending on the task, to exit earlier
- So we can imagine to have a distribution for each token and layer



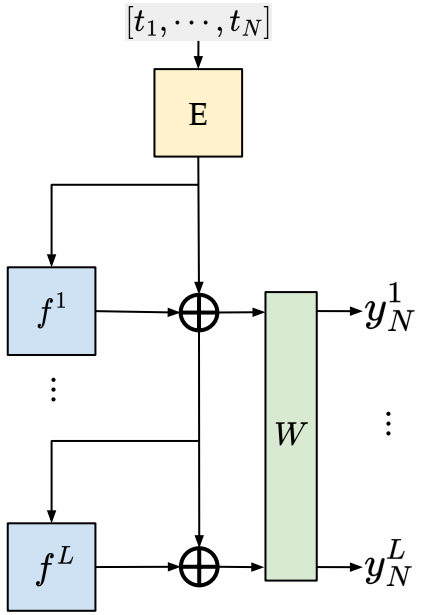
Motivation: Early-exit

- These distributions are:



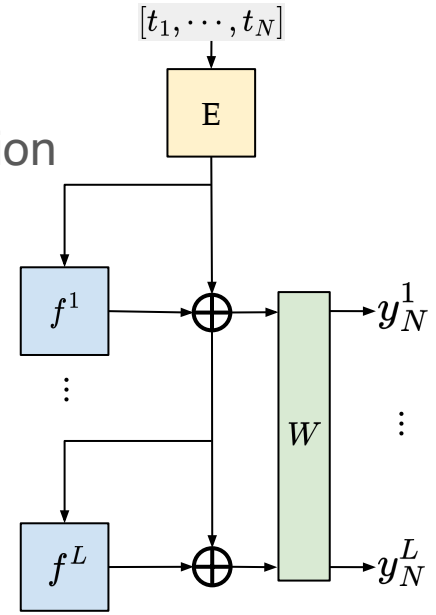
Motivation: Early-exit

- These distributions are:
    - High dimensional



Motivation: Early-exit

- These distributions are:
    - High dimensional → Dimensionality reduction

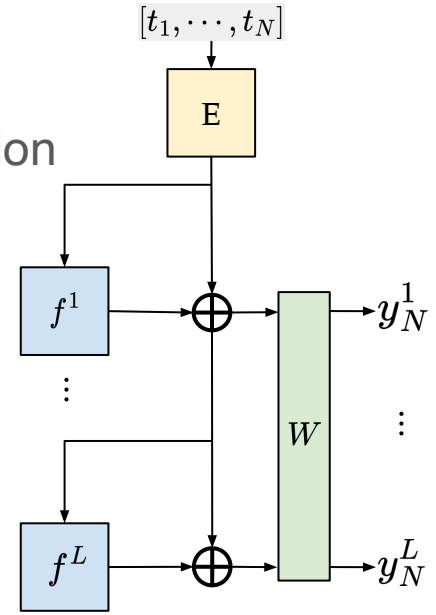


Motivation: Early-exit

- These distributions are:
  - High dimensional → Dimensionality reduction

↓

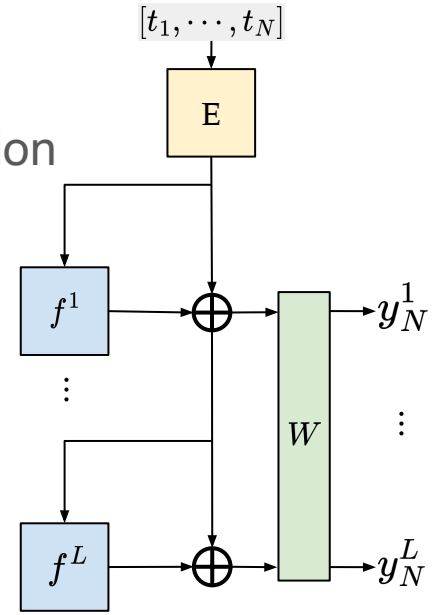
Moments, cumulants



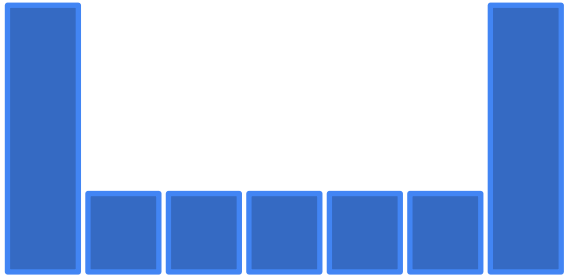
Motivation: Early-exit

- These distributions are:
  - High dimensional → Dimensionality reduction

Moments, ~~cumulants~~

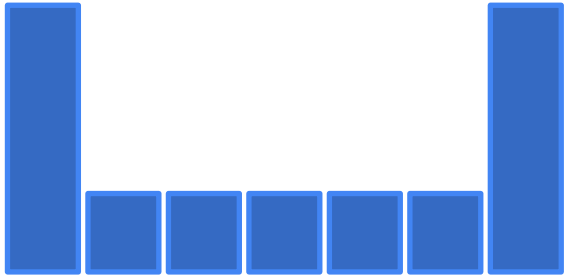


Why moments and cumulants don’t work anymore



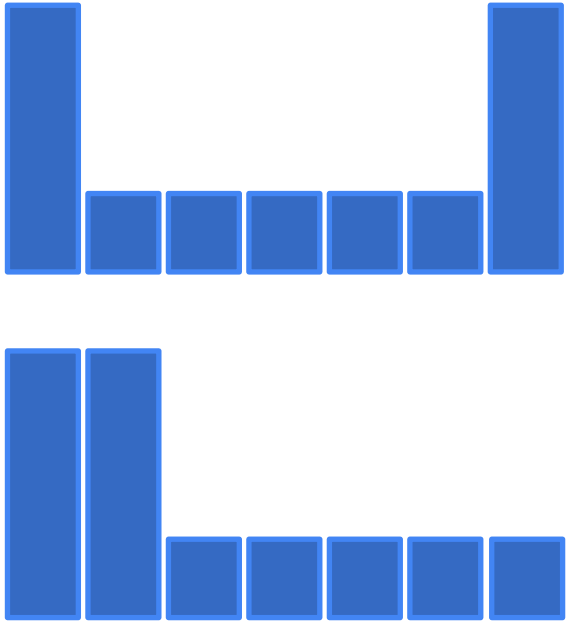
Why moments and cumulants don’t work anymore

- High variance



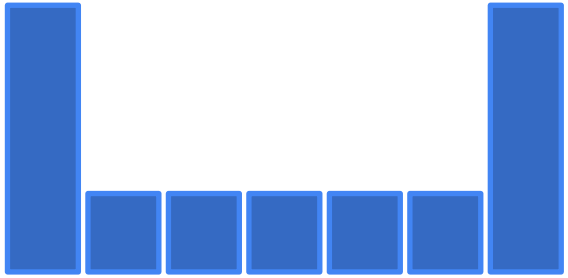
Why moments and cumulants don’t work anymore

- High variance

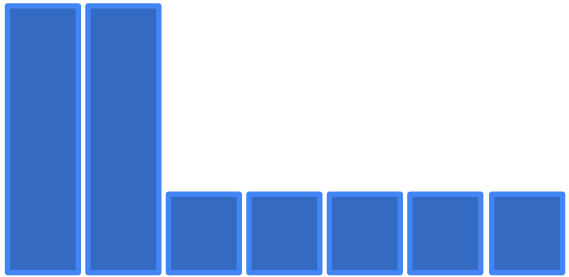


Why moments and cumulants don’t work anymore

- High variance



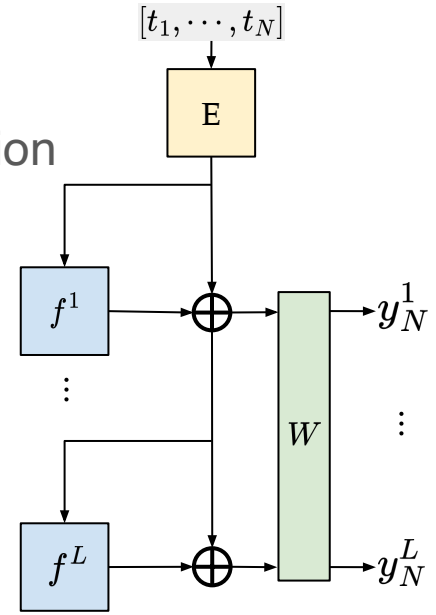
- Low variance



Motivation: Early-exit

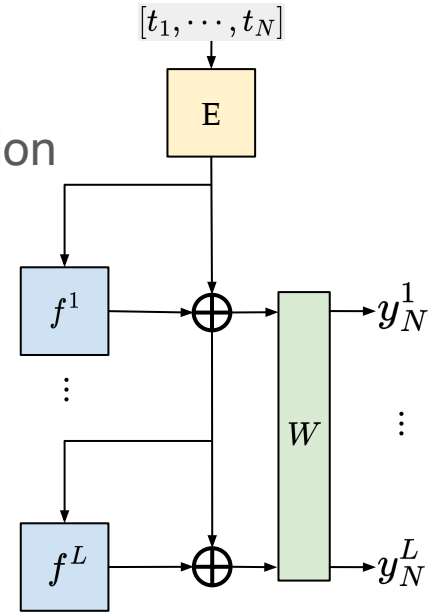
- These distributions are:
  - High dimensional → Dimensionality reduction

Moments, ~~cumulants~~



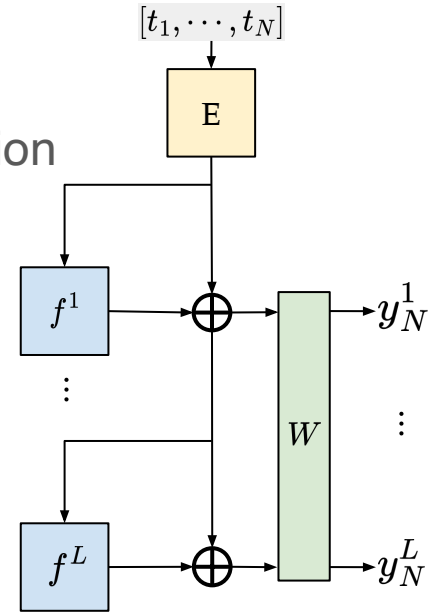
Motivation: Early-exit

- These distributions are:
  - High dimensional → Dimensionality reduction
  - Unordered support → (Rényi) Entropy



Motivation: Early-exit

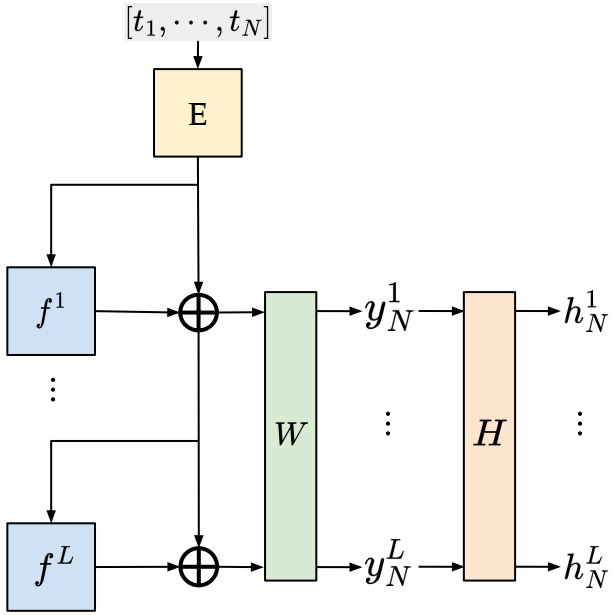
- These distributions are:
  - High dimensional → Dimensionality reduction
  - Unordered support → (Rényi) Entropy



- DISCLAIMER: we'll show experimentally that we can use Shannon entropy instead of Rényi one.

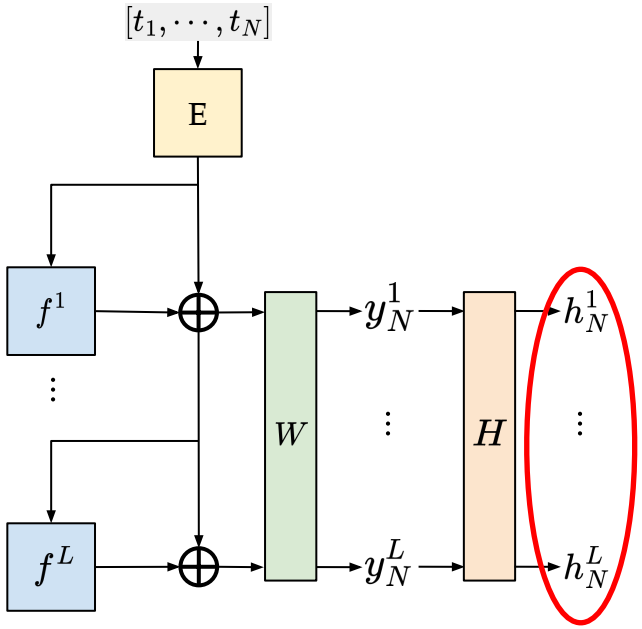
Our framework: Entropy profiles

- We look at the evolution of the probability distribution through entropy



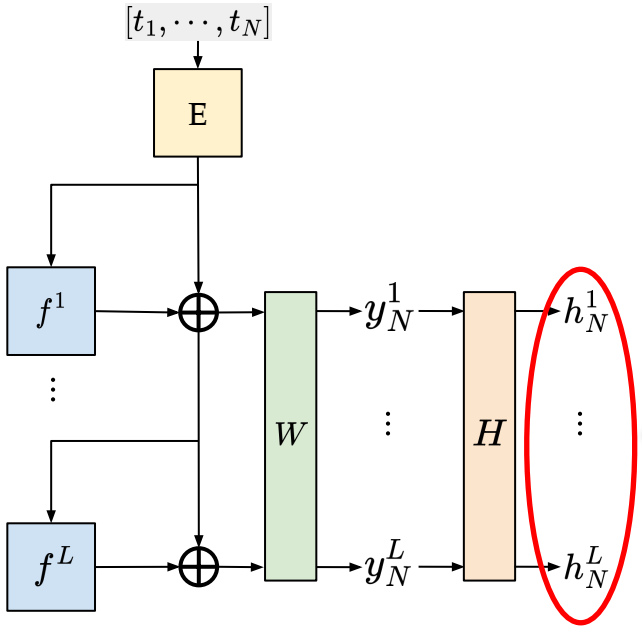
Our framework: Entropy profiles

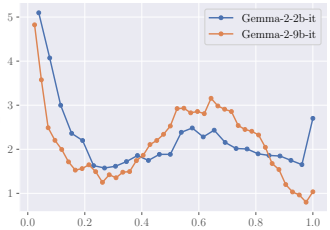
- We look at the evolution of the probability distribution through entropy



Our framework: Entropy profiles

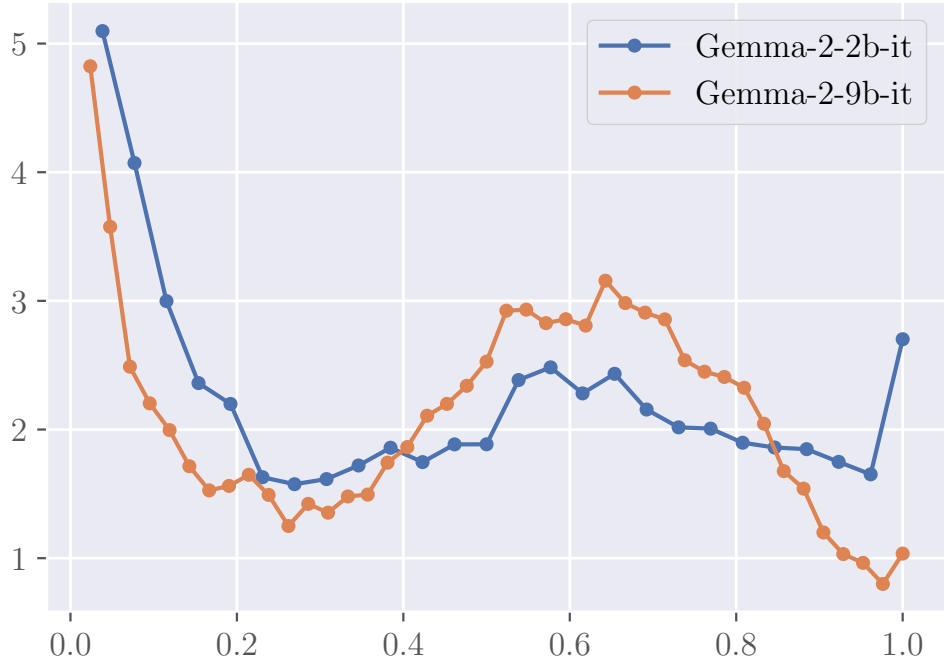
- We look at the evolution of the probability distribution through entropy





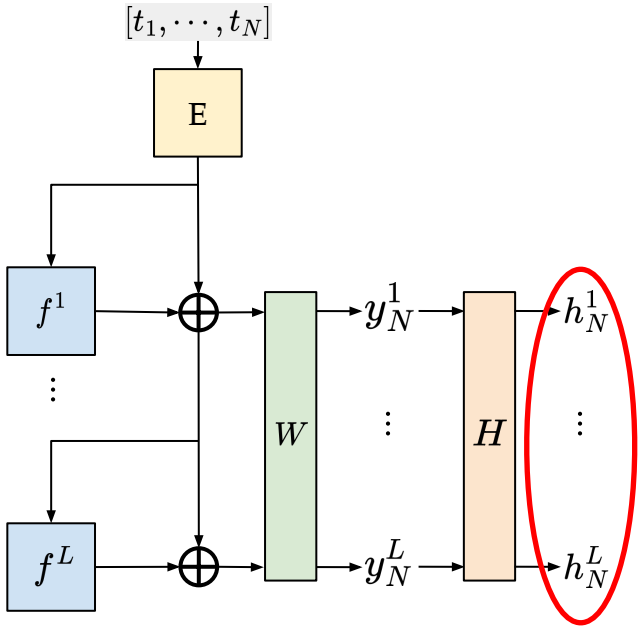
Our framework: Entropy profiles

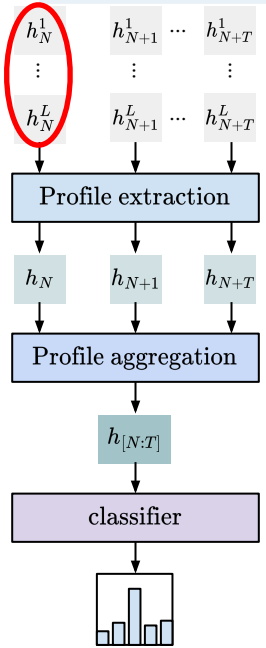
- We look at the evolution of the probability distribution through entropy



Our framework: Entropy profiles

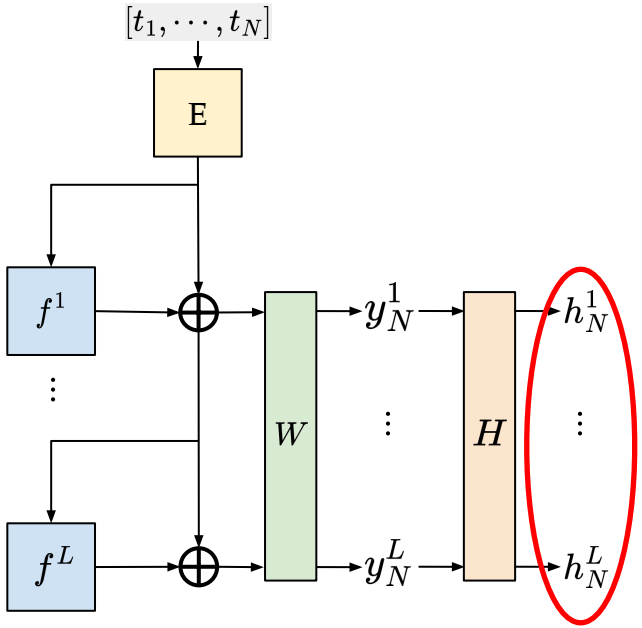
- We look at the evolution of the probability distribution through entropy
- We aggregate the entropy profiles from different tokens

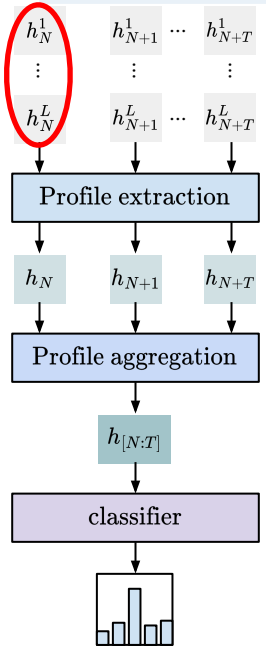




Our framework: Entropy profiles

- We look at the evolution of the probability distribution through entropy
- We aggregate the entropy profiles from different tokens
- We study which kind of information they retain





Can entropy profiles identify models?

Motivation: Graphs encode structure at multiple levels

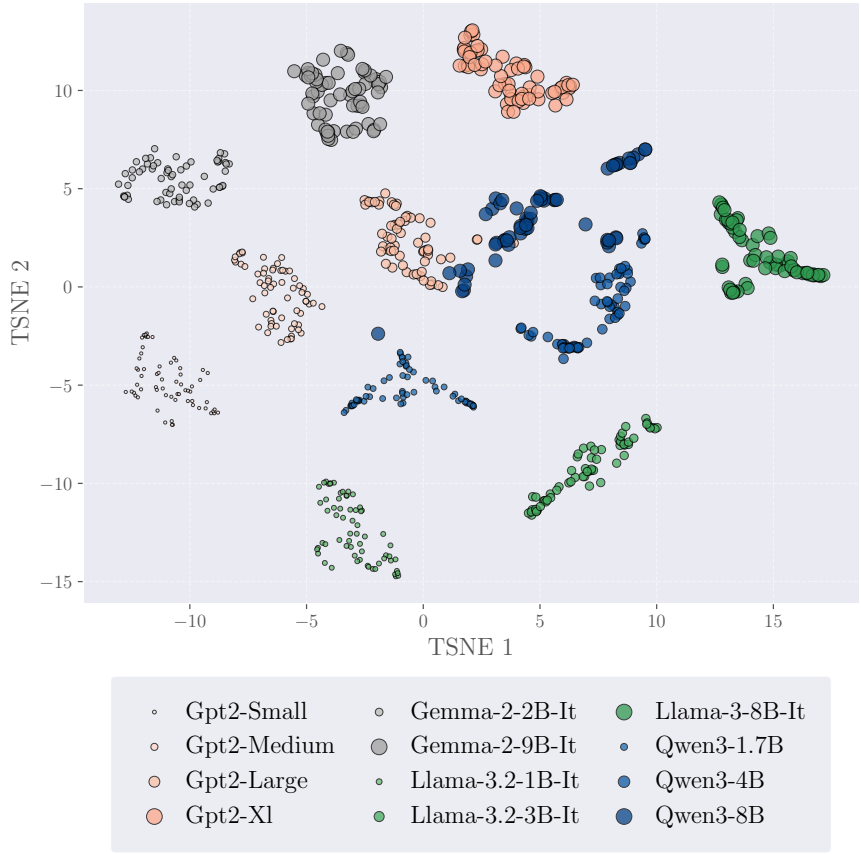
- 12 models, 4 families (GPT, LLaMA, Gemma, Qwen).

Motivation: Graphs encode structure at multiple levels

- 12 models, 4 families (GPT, LLaMA, Gemma, Qwen).
- t-SNE clusters by family, independent of model size.

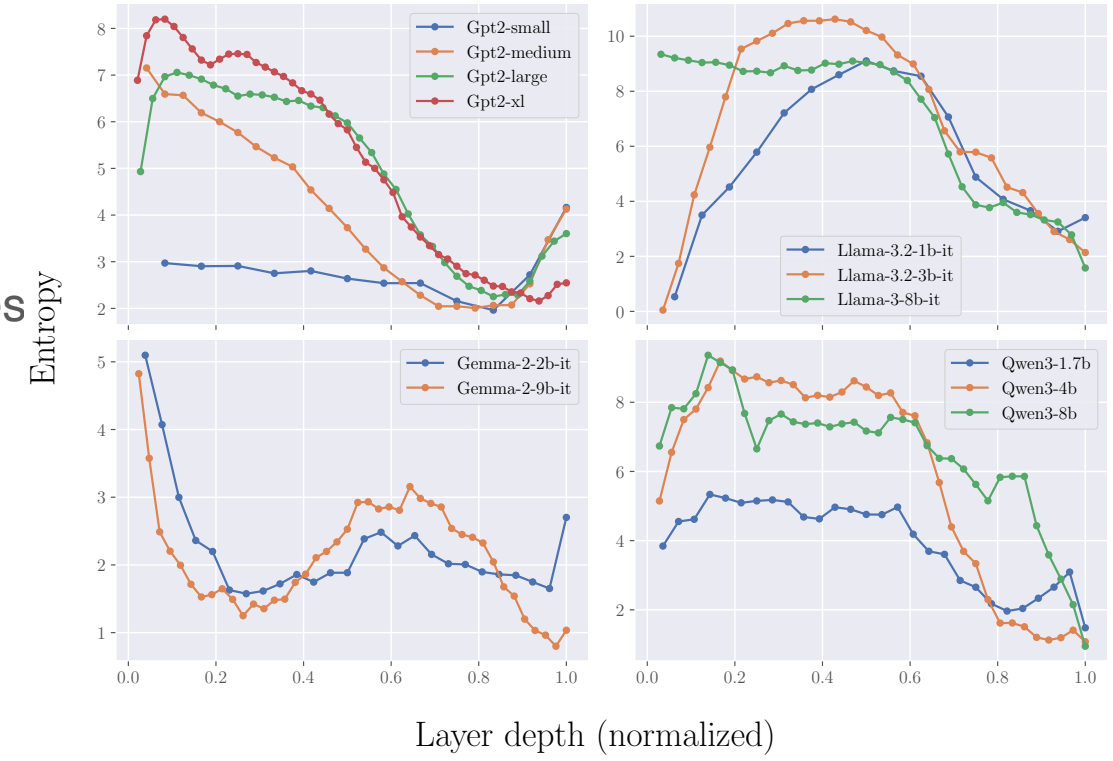
Motivation: Graphs encode structure at multiple levels

- 12 models, 4 families (GPT, LLaMA, Gemma, Qwen).
- t-SNE clusters by family, independent of model size.

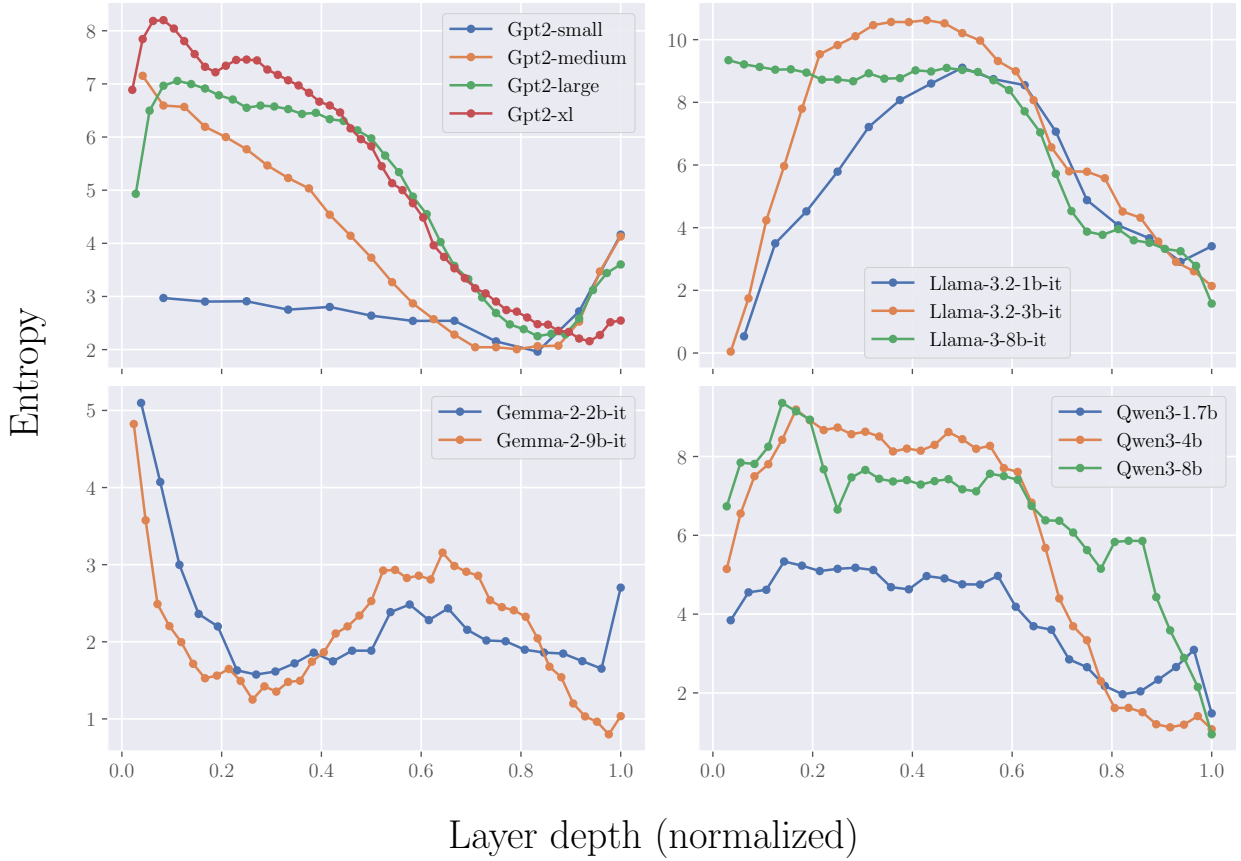


Motivation: Graphs encode structure at multiple levels

- 12 models, 4 families (GPT, LLaMA, Gemma, Qwen).
- t-SNE clusters by family, independent of model size.
- After depth normalisation, shapes are invariant.



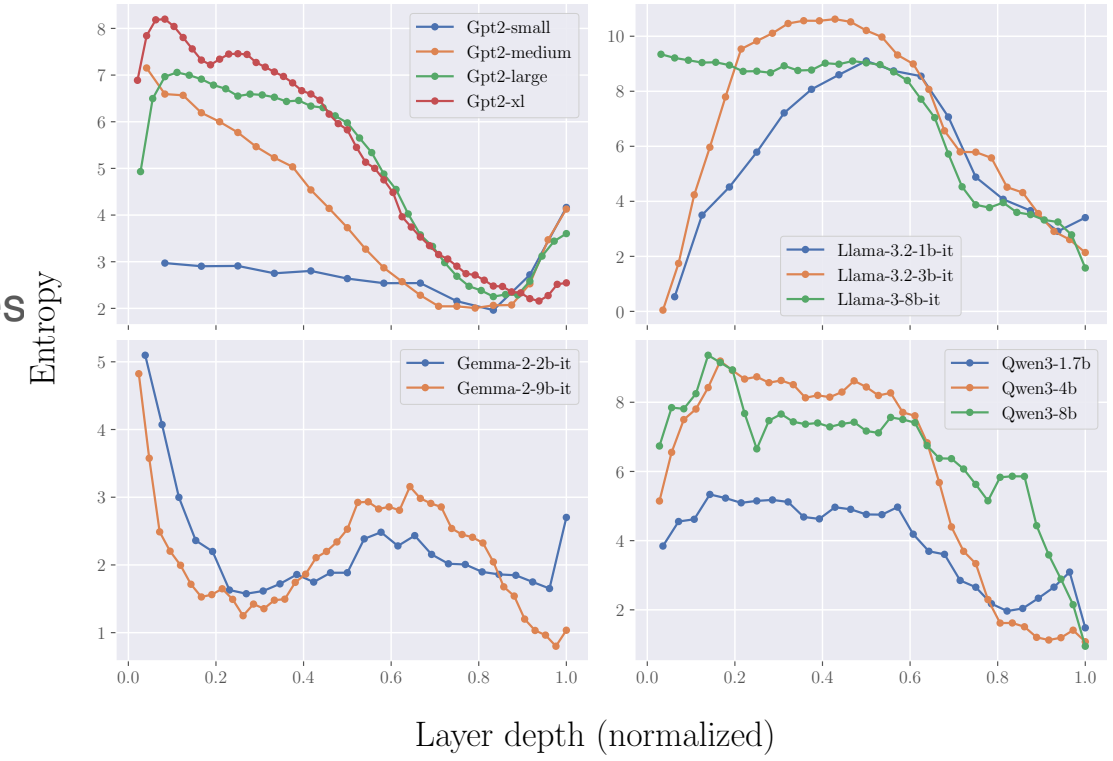
Motivation: Graphs encode structure at multiple levels



Motivation: Graphs encode structure at multiple levels

- 12 models, 4 families (GPT, LLaMA, Gemma, Qwen).
- t-SNE clusters by family, independent of model size.
- After depth normalisation, shapes are invariant.

Task	Macro F1-score
model family	97.99±0.66
model size	96.31±0.87



Can entropy profiles identify the task?

Results: Task differentiation

- Dataset: *TinyStories* → generative, syntactic, semantic tasks.

Results: Task differentiation

- Dataset: *TinyStories* → generative, syntactic, semantic tasks.
- Three prompt templates: Base, Reversed, and Scrambled
- Total of 2400 prompts

Results: Task differentiation

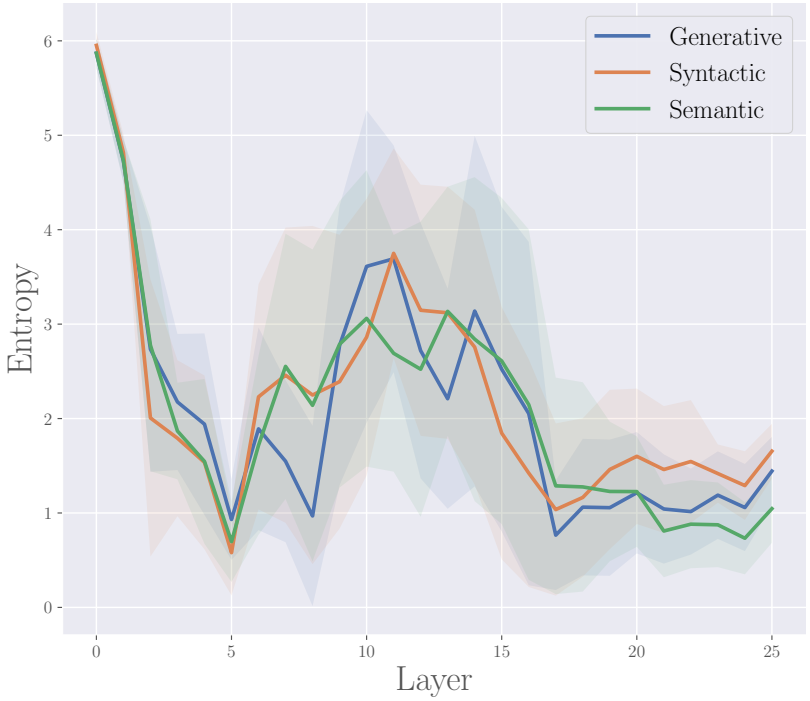
- Dataset: *TinyStories* → generative, syntactic, semantic tasks.
- Three prompt templates: Base, Reversed, and Scrambled
- Total of 2400 prompts
- 10-fold cross validation ROC-AUC (1-vs-rest)

Results: Task differentiation

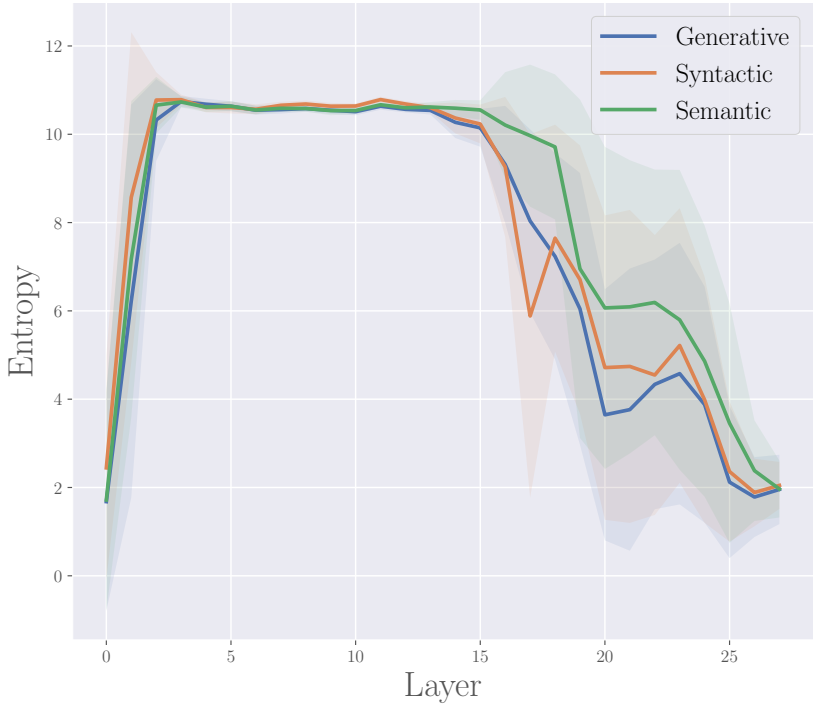
Model	Size	k-NN AUC
Gemma-2-it	2.1B	97.66 ± 0.47
Gemma-2-it	8.9B	98.38 ± 0.50
Llama-3.2-it	1B	94.94 ± 0.79
Llama-3.2-it	3B	94.77 ± 0.93
Llama-3-it	8B	96.10 ± 0.67
Phi-3	3.6B	97.07 ± 0.87

Results: Task differentiation

Llama-3.2-it



Gemma-2-it



Results: Task differentiation

Considered layers	k-NN AUC
`first-only`	68.34 ± 2.68
`middle-only`	78.83 ± 3.07
`last-only`	76.78 ± 2.36
`first+middle+last`	86.13 ± 1.41
`all`	90.49 ± 1.76

Do entropy profiles correlate with correct task execution?

Results: Correct execution

- Dataset: *MMLU* —> correct, incorrect.

Results: Correct execution

- Dataset: *MMLU* —> correct, incorrect.
- Three prompt templates: Base, Instruct, and Humble

Results: Correct execution

- Dataset: *MMLU* —> correct, incorrect.
- Three prompt templates: Base, Instruct, and Humble
- 10-fold cross validation ROC-AUC (1-vs-rest)

Results: Correct execution

- Dataset: *MMLU* —> correct, incorrect.
- Three prompt templates: Base, Instruct, and Humble
- 10-fold cross validation ROC-AUC (1-vs-rest)
- Dummy model: sampled from distribution reflecting the proportion of correct and incorrect answers.

Results: Correct execution

Model	Prompt	LLM-Acc.	k-NN AUC
Llama	Base	50.89	73.61 ± 1.52
	Humble	58.51	69.90 ± 1.06
	Instruct	60.62	67.23 ± 1.62
Gemma	Base	56.10	71.88 ± 1.63
	Humble	54.71	72.78 ± 1.15
	Instruct	56.38	68.36 ± 1.23

Is Shannon entropy a good choice?

Shannon entropy

- For a discrete random variable X with output x_i and probability mass function p

Shannon entropy

- For a discrete random variable X with output x_i and probability mass function p
- Shannon Entropy $H(X) = -\sum_i p(x_i)\log p(x_i)$

Rényi entropy

- For a discrete random variable X with output x_i and probability mass function p.

Rényi entropy

- For a discrete random variable X with output x_i and probability mass function p.
- Rényi Entropy $H_\alpha(X) = \frac{1}{1-\alpha} \log \sum_i p(x_i)^\alpha$ for $\alpha > 0, \alpha \neq 1.$

Rényi entropy

- For a discrete random variable X with output x_i and probability mass function p.
- Rényi Entropy $H_\alpha(X) = \frac{1}{1-\alpha} \log \sum_i p(x_i)^\alpha$ for $\alpha > 0, \alpha \neq 1.$
- α is (very) roughly speaking like a temperature

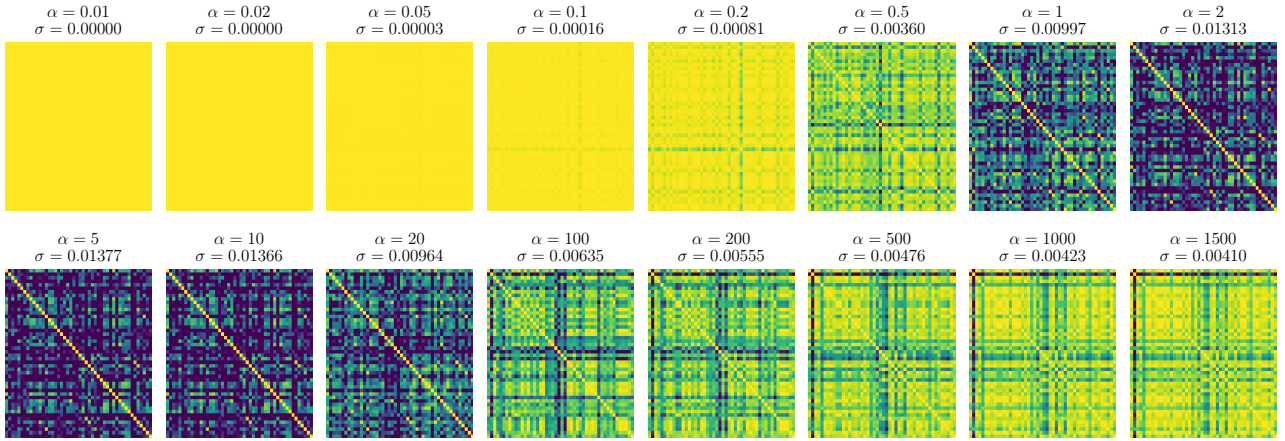
Rényi entropy

- For a discrete random variable X with output x_i and probability mass function p.
- Rényi Entropy $H_\alpha(X) = \frac{1}{1-\alpha} \log \sum_i p(x_i)^\alpha$ for $\alpha > 0, \alpha \neq 1$.
- It reduce to Shannon entropy in the limit $\alpha \to 1$.

Rényi entropy

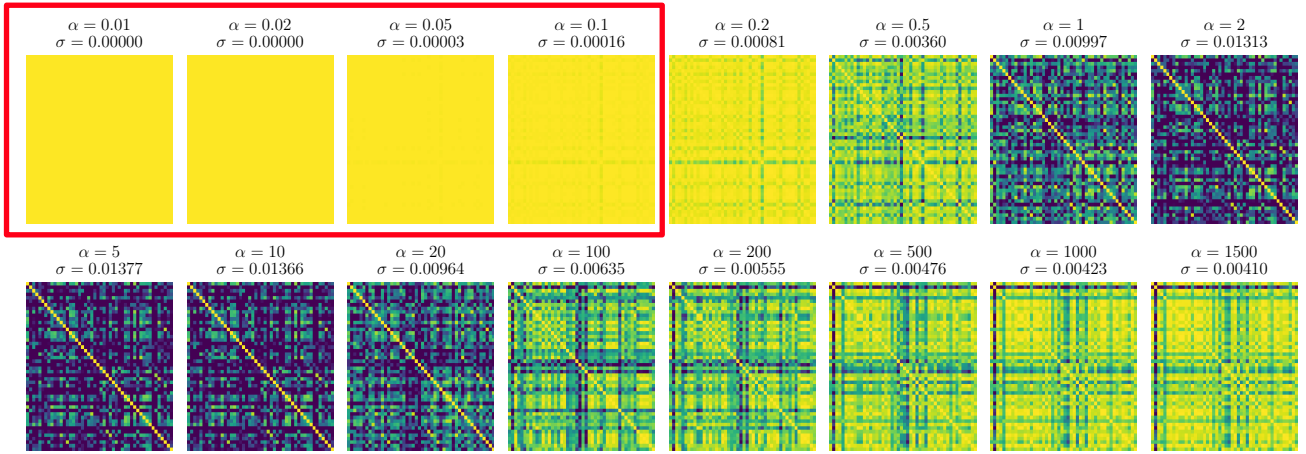
- For a discrete random variable X with output x_i and probability mass function p.
- Rényi Entropy $H_\alpha(X) = \frac{1}{1-\alpha} \log \sum_i p(x_i)^\alpha$ for $\alpha > 0$, $\alpha \neq 1$.
- It reduce to Shannon entropy in the limit $\alpha \to 1$.
- And to other permutation invariant measures: collision entropy ($\alpha = 2$), min-entropy ($\alpha \to \infty$), max-entropy ($\alpha \to 0$)…
- It correlates with indexes like the Gini-Simpson one.

The three regimes of Rényi entropy

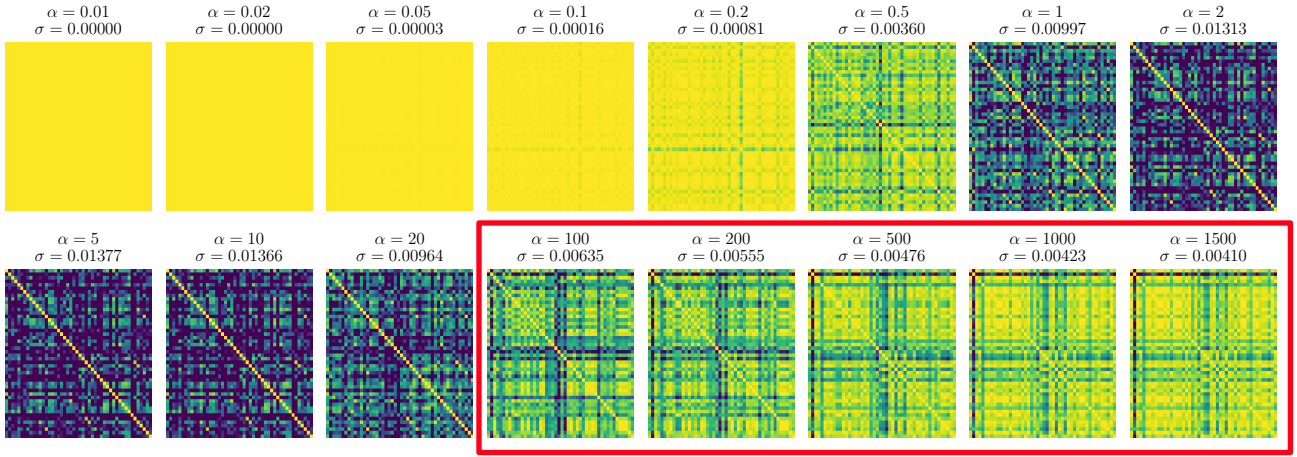


The three regimes of Rényi entropy

α saturates, all tokens are considered



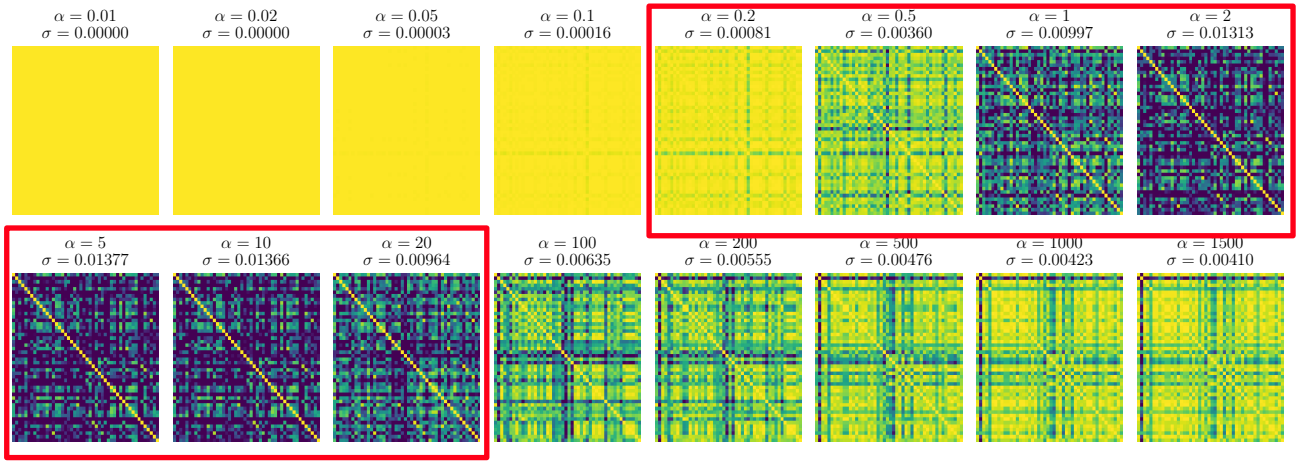
The three regimes of Rényi entropy



The profile collapse into considering only the most probable token

The three regimes of Rényi entropy

Informative regime



It contains Shannon entropy

Can entropy profiles identify text format?

Results: Text format

- Dataset: *custom → poem, scientific piece, and chat log. (Topic-format dataset)*

Results: Text format

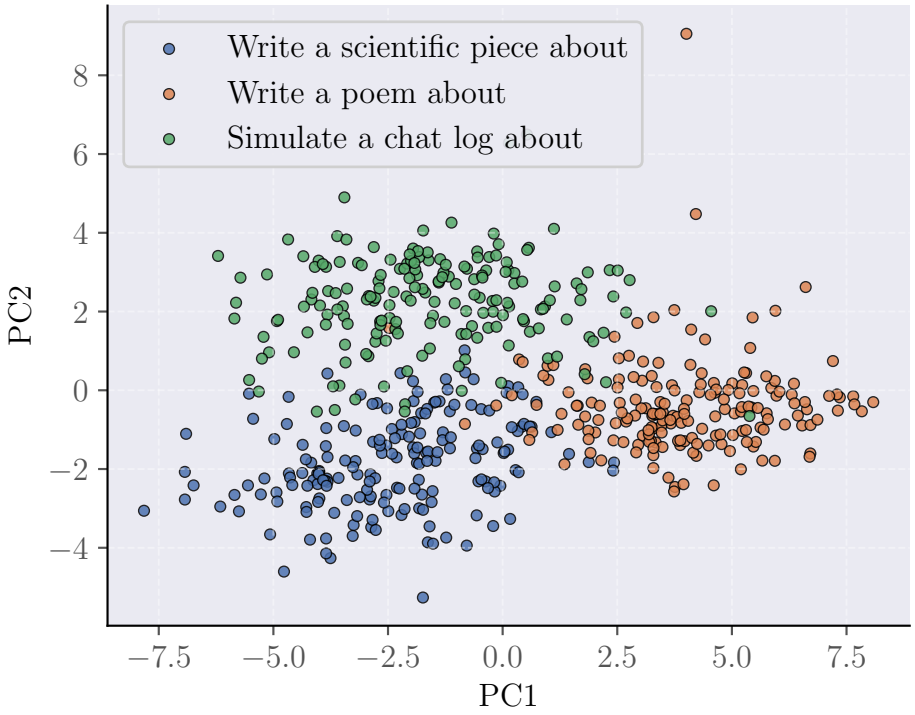
- Dataset: *custom → poem, scientific piece, and chat log. (Topic-format dataset)*
- Across different α values in the informative range.

Results: Text format

- Dataset: *custom → poem, scientific piece, and chat log. (Topic-format dataset)*
- Across different α values in the informative range.

Model	α	k-NN AUC
Gemma-2-2B-it	0.5	97.3 ± 1.6
	1.0	98.7 ± 1.1
	5.0	98.4 ± 1.7
Llama-3.2-1B-it	0.5	97.8 ± 1.6
	1.0	97.8 ± 2.4
	5.0	96.6 ± 2.6

Results: Text format



Can we give a theoretical explanation?

Memorization: From Shannon

- Training data distribution X, data generating processes distribution Θ (the law), (training algorithm $L : X \to \hat{\Theta}$,) trained models distribution $\hat{\Theta}$.

Brown et al.

Memorization: From Shannon

- Training data distribution X, data generating processes distribution Θ (the law), (training algorithm $L : X \to \hat{\Theta}$,) trained models distribution $\hat{\Theta}$.
- $\text{mem}(X, \hat{\Theta}) = I(X, \hat{\Theta}) = H(X) - H(X \mid \hat{\Theta})$

Brown et al.

Memorization: From Shannon

- Training data distribution X, data generating processes distribution Θ (the law), (training algorithm $L : X \to \hat{\Theta}$,) trained models distribution $\hat{\Theta}$.
- $\text{mem}(X, \hat{\Theta}) = I(X, \hat{\Theta}) = H(X) - H(X \mid \hat{\Theta})$

How much info is in X

Brown et al.

Memorization: From Shannon

- Training data distribution X, data generating processes distribution Θ (the law), (training algorithm $L : X \to \hat{\Theta}$,) trained models distribution $\hat{\Theta}$.
- $\text{mem}(X, \hat{\Theta}) = I(X, \hat{\Theta}) = H(X) - H(X \mid \hat{\Theta})$

How much info is in X

How much info is in X
BUT NOT in $\hat{\Theta}$

Brown et al.

Memorization: To Kolmogorov

- Training data instance x, data generating process θ (the law), (training algorithm $L : x \to \hat{\theta}$,) trained model $\hat{\theta}$.

Morris et al.

Memorization: To Kolmogorov

- Training data instance x, data generating process θ (the law), (training algorithm $L : x \to \hat{\theta}$,) trained model $\hat{\theta}$.
- Kolmogorov complexity of an instance x given model parameters $\hat{\theta}$ is

$$H^k(x \mid \hat{\theta}) = \min_s \{ \mid s \mid : f(s, \hat{\theta}) = x \}$$

Morris et al.

Memorization: To Kolmogorov

- Training data instance x, data generating process θ (the law), (training algorithm $L : x \to \hat{\theta}$,) trained model $\hat{\theta}$.
- Kolmogorov complexity of an instance x given model parameters $\hat{\theta}$ is

$$H^k(x \mid \hat{\theta}) = \min_s \{ \mid s \mid : f(s, \hat{\theta}) = x \}$$

- $H^k(x \mid \hat{\theta}) \approx -\log p(x \mid \hat{\theta})$

Morris et al.

Memorization: To Kolmogorov

- From Grunwald & Vitányi (2004)

$$I(X, Y) - H^K(f) \leq \mathbb{E}_{(x,y)\sim(X,Y)}[I^K(x, y)] \leq I(X, Y) + 2H^K(f)$$

Memorization: To Kolmogorov

- From Grunwald & Vitányi (2004)

$$I(X,Y) - H^K(f) \leq \mathbb{E}_{(x,y)\sim(X,Y)}[I^K(x,y)] \leq I(X,Y) + 2H^K(f)$$

- Remember $I^K(x,x) = H^K(x)$; then

Memorization: To Kolmogorov

- From Grunwald & Vitányi (2004)

$$I(X,Y) - H^K(f) \leq \mathbb{E}_{(x,y)\sim(X,Y)}[I^K(x,y)] \leq I(X,Y) + 2H^K(f)$$

- Remember $I^K(x,x) = H^K(x)$; then
- $$H^K(X \mid \hat{\Theta} = \hat{\theta}) - H^K(f) \leq \mathbb{E}_{(x\mid\hat{\theta})\sim(X\mid\hat{\Theta}=\hat{\theta})}[H^K(x \mid \hat{\theta})] \leq H(X \mid \hat{\Theta} = \hat{\theta}) + 2H^K(f)$$

Memorization: To Entropy profiles

- $H^k(x \mid \hat{\theta}) \approx -\log p(x \mid \hat{\theta})$

Memorization: To Entropy profiles

- $H^k(x \mid \hat{\theta}) \approx -\log p(x \mid \hat{\theta})$
- $H(X \mid \hat{\Theta} = \hat{\theta}) - H^K(f) \leq \mathbb{E}_{(x|\hat{\theta}) \sim (X|\hat{\Theta}=\hat{\theta})}[H^K(x \mid \hat{\theta})] \leq H(X \mid \hat{\Theta} = \hat{\theta}) + 2H^K(f)$

Memorization: To Entropy profiles

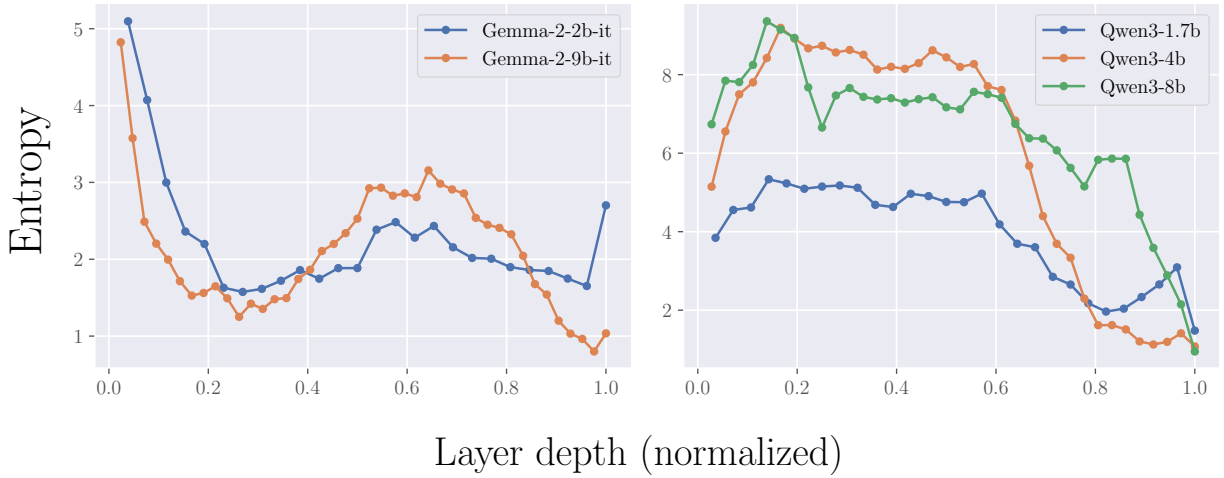
- $H^k(x \mid \hat{\theta}) \approx -\log p(x \mid \hat{\theta})$
- $H(X \mid \hat{\Theta} = \hat{\theta}) - H^K(f) \leq \mathbb{E}_{(x|\hat{\theta}) \sim (X|\hat{\Theta}=\hat{\theta})}[H^K(x \mid \hat{\theta})] \leq H(X \mid \hat{\Theta} = \hat{\theta}) + 2H^K(f)$
- $H(X \mid \hat{\Theta} = \hat{\theta}) = H(X) - I(X \mid \hat{\Theta} = \hat{\theta}) = H(X) - \mathrm{mem}(X, \hat{\Theta} = \hat{\theta})$

Memorization: To Entropy profiles

- Morris et al. analysed memorisation for the whole model

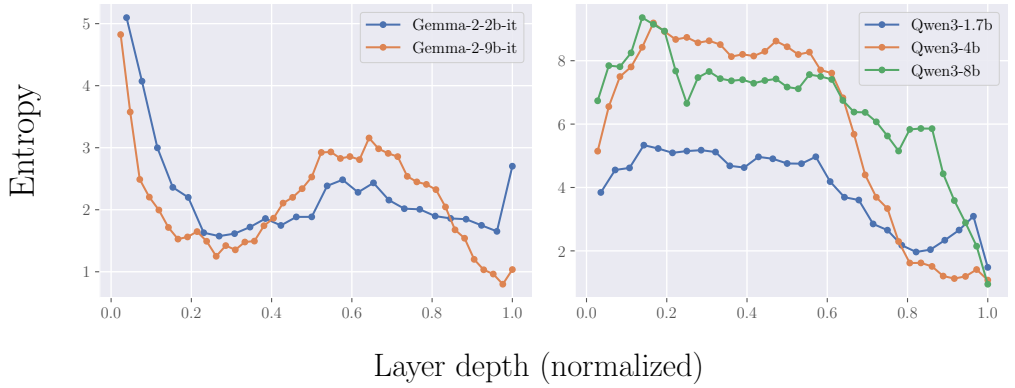
Memorization: To Entropy profiles

- Morris et al. analysed memorisation for the whole model
- We showed memorisation for sub-models composed of the first n-layers



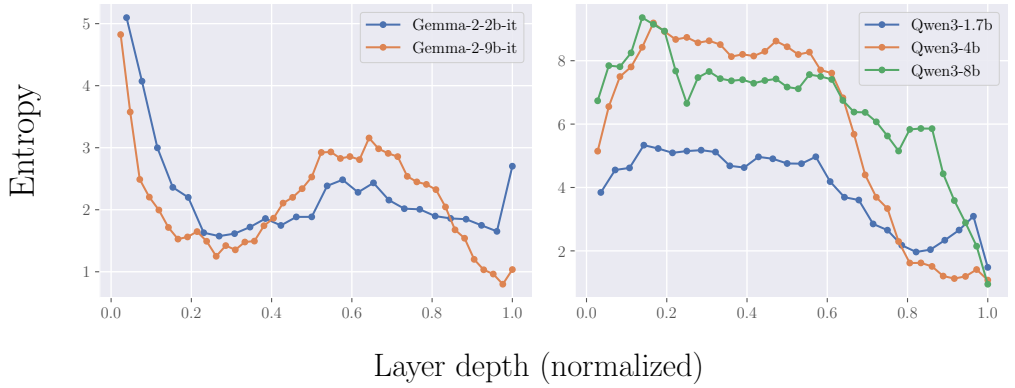
Memorization: To Entropy profiles

- Morris et al. analysed memorisation for the whole model
- We showed memorisation for sub-models composed of the first n-layers
- Surprisingly the growth is non-monotonic

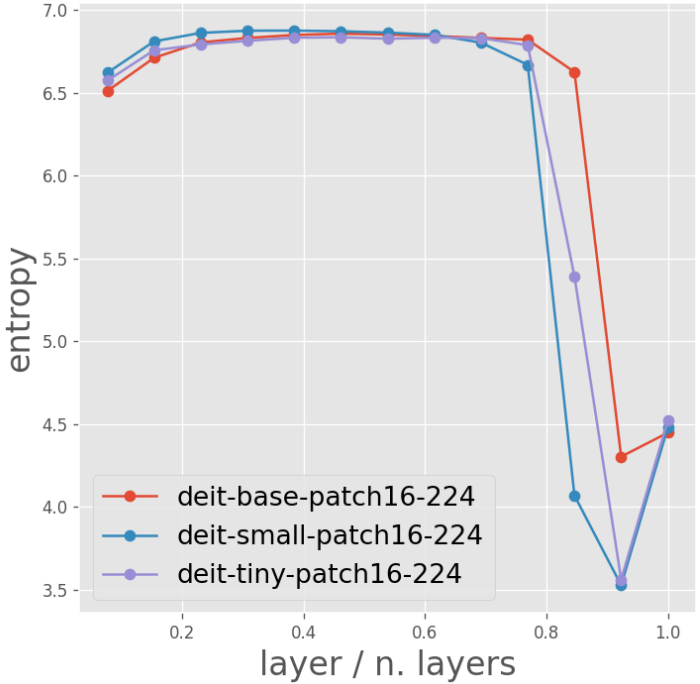


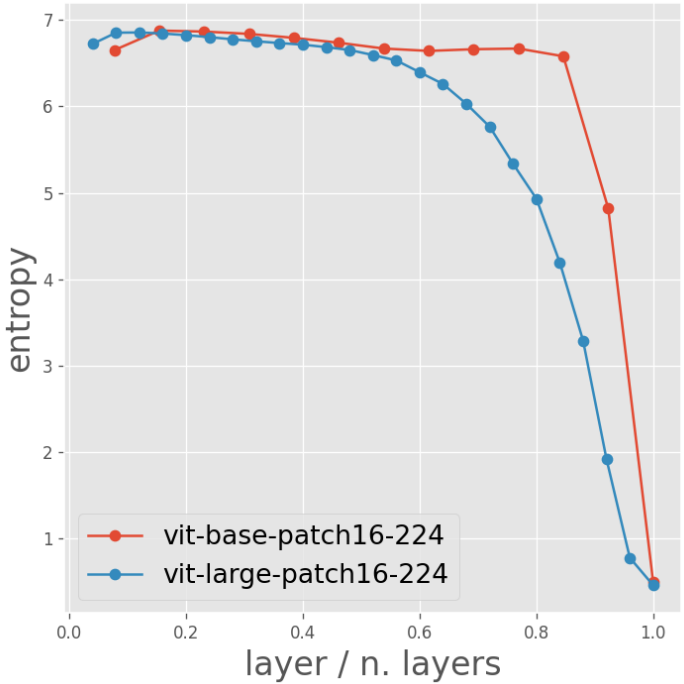
Memorization: To Entropy profiles

- Morris et al. analysed memorisation for the whole model
- We showed memorisation for sub-models composed of the first n-layers
- Surprisingly the growth is non-monotonic
- It depends on family (architecture), task, and format.



Beyond language: Vision Transformers





Limitations and Future Directions

- Limitations
- The idea of **task** is not well defined.
- The idea of **format** is not well defined.
- We don’t know **why families show characteristic entropy profiles**.

- Future Work
- Understand **which parts of the architecture influence the entropy profile**.
- Further connect it to **memorisation**.

Conclusions

- Can we study how the probability of tokens evolve through the layers of a Transformer? **Yes, through their entropy.**
- What kind of information do entropy profiles contain? **They contain information on which task is being computed, which kind of format is being produced and, surprisingly, the family and model processing the information.**
- Can we get a grasp on what these profile represent? **These profiles tell us that different models memorise differently, also depending on task and format, and most importantly that their memorisation doesn't grow with the number of parameters.**

PAPER

CODE

Summary of links

CONTACTS

Renormalization Group

NNs as RG flow

PAPER

RG for GNNs

PAPER

CODE

Entropy

MaxEnt for circuits

PAPER AND CODE

Entropy for Transformers

PAPER

CODE