

MaNo: Exploiting Matrix Norm for Unsupervised Accuracy Estimation

Ambroise Odonnat

Noah's Ark Lab, Inria
Université Rennes 2, CNRS, IRISA

Obelix Team, IRISA

February 6, 2025



Inria





Co-authors



Renchunzi Xie
NTU



Vasilii Feofanov
Noah's Ark Lab



Weijian Deng
ANU



Jianfeng Zhang
Noah's Ark Lab



Bo An
NTU



Outline

- ① Introduction
- ② First Principle Analysis
- ③ Our Method: MaNo
- ④ Experimental Results
- ⑤ Take Home Message



Outline

① Introduction

② First Principle Analysis

③ Our Method: MaNo

④ Experimental Results

⑤ Take Home Message



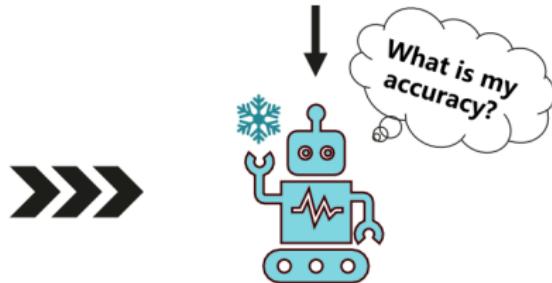
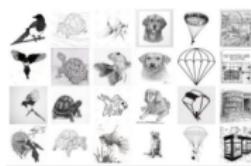
Unsupervised Accuracy Estimation

Goal: Predict accuracy of pre-trained model f on test set $\mathcal{D}_{\text{test}}$.

Labeled Training Data



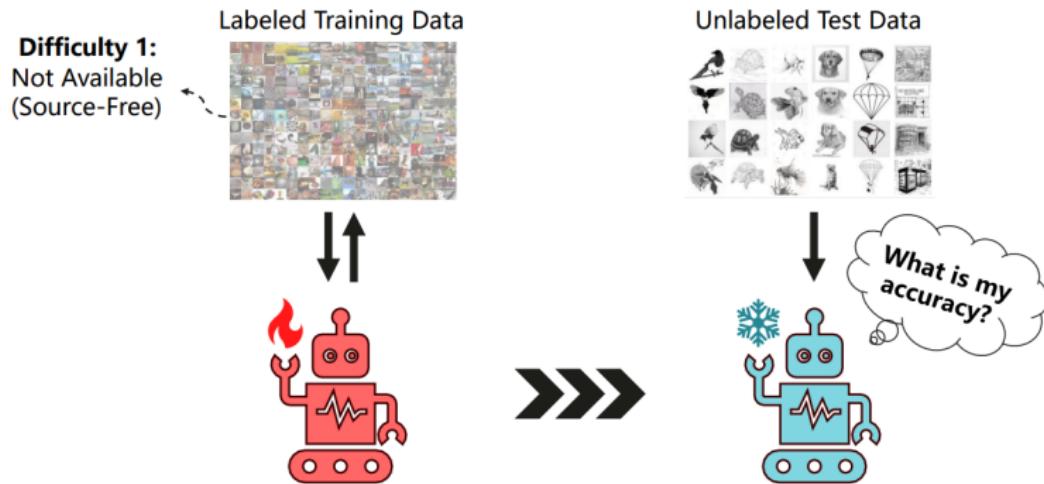
Unlabeled Test Data





Unsupervised Accuracy Estimation

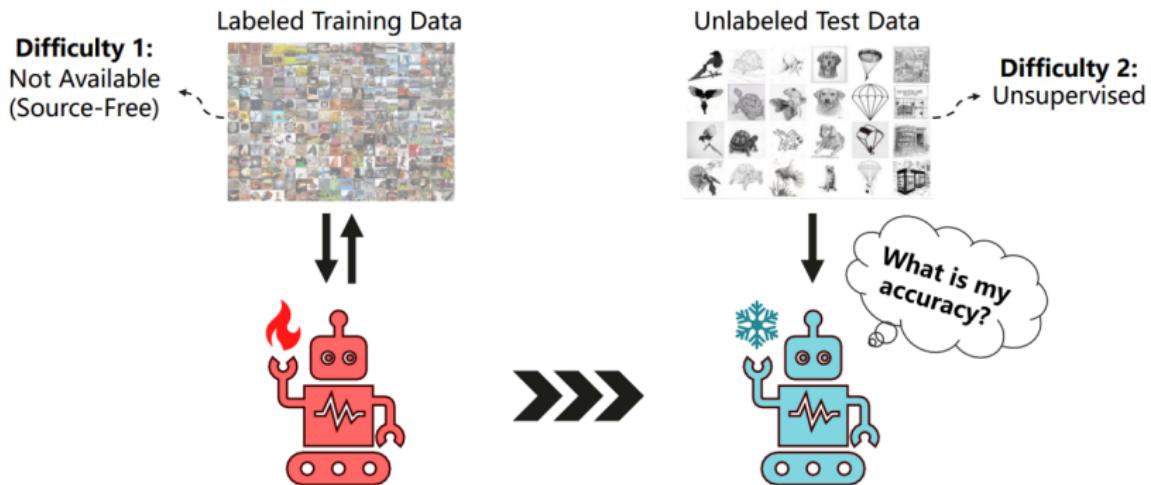
Goal: Predict accuracy of pre-trained model f on test set $\mathcal{D}_{\text{test}}$.





Unsupervised Accuracy Estimation

Goal: Predict accuracy of pre-trained model f on test set $\mathcal{D}_{\text{test}}$.

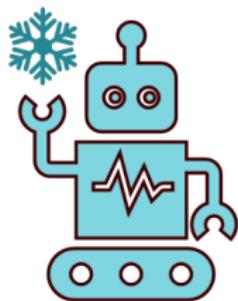
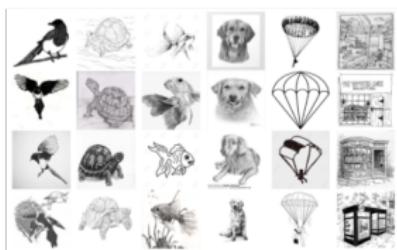


→ Challenging task often occurring in real-world scenarios.



Logits-based Methods

Unlabeled Test Data



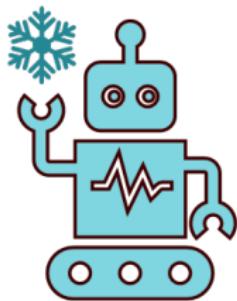


Logits-based Methods

Unlabeled Test Data



✓ Model's outputs: logits



logits: $\mathbf{q}_i = (\mathbf{w}_1^\top \phi(\mathbf{x}_i), \dots, \mathbf{w}_K^\top \phi(\mathbf{x}_i))$,

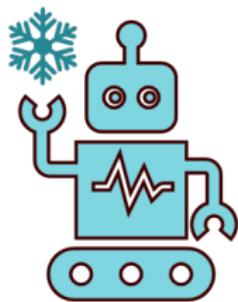


Logits-based Methods

Unlabeled Test Data



- ✓ Model's outputs: logits
- ✓ Different range → normalize



logits: $\mathbf{q}_i = (\mathbf{w}_1^\top \phi(\mathbf{x}_i), \dots, \mathbf{w}_K^\top \phi(\mathbf{x}_i))$,
normalizer: $\sigma : \mathbb{R}^K \rightarrow \Delta_K$.

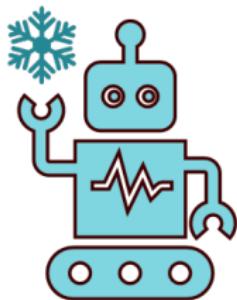


Logits-based Methods

Unlabeled Test Data



- ✓ Model's outputs: logits
- ✓ Different range → normalize
- ✓ Fill prediction matrix \mathbf{Q}



logits: $\mathbf{q}_i = (\mathbf{w}_1^\top \phi(\mathbf{x}_i), \dots, \mathbf{w}_K^\top \phi(\mathbf{x}_i))$,
normalizer: $\sigma : \mathbb{R}^K \rightarrow \Delta_K$.

$$\rightarrow \mathbf{Q} = \begin{pmatrix} \sigma(\mathbf{q}_1) \\ \vdots \\ \sigma(\mathbf{q}_N) \end{pmatrix}$$

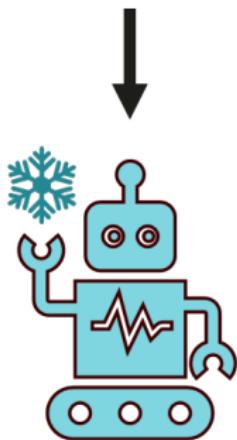


Logits-based Methods

Unlabeled Test Data



- ✓ Model's outputs: logits
- ✓ Different range → normalize
- ✓ Fill prediction matrix \mathbf{Q}
- ✓ Compute estimation score



logits: $\mathbf{q}_i = (\mathbf{w}_1^\top \phi(\mathbf{x}_i), \dots, \mathbf{w}_K^\top \phi(\mathbf{x}_i))$,
normalizer: $\sigma : \mathbb{R}^K \rightarrow \Delta_K$.

$$\rightarrow \mathbf{Q} = \begin{pmatrix} \sigma(\mathbf{q}_1) \\ \vdots \\ \sigma(\mathbf{q}_N) \end{pmatrix} \rightarrow \text{Score}$$



Research Questions

Question 1: *What explains the correlation between logits and generalization performance?*



Research Questions

Question 1: *What explains the correlation between logits and generalization performance?*

Question 2: *How to alleviate the overconfidence issues of logits-based methods?*



Outline

① Introduction

② First Principle Analysis

③ Our Method: MaNo

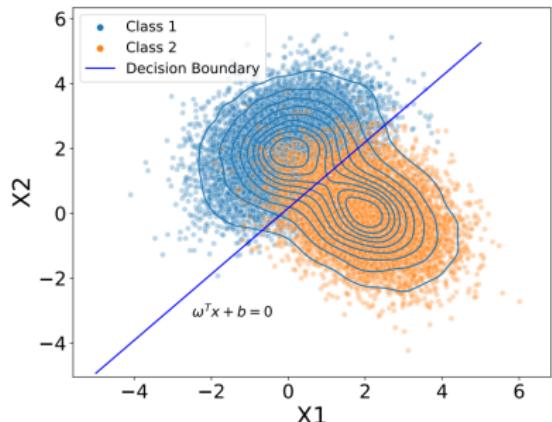
④ Experimental Results

⑤ Take Home Message

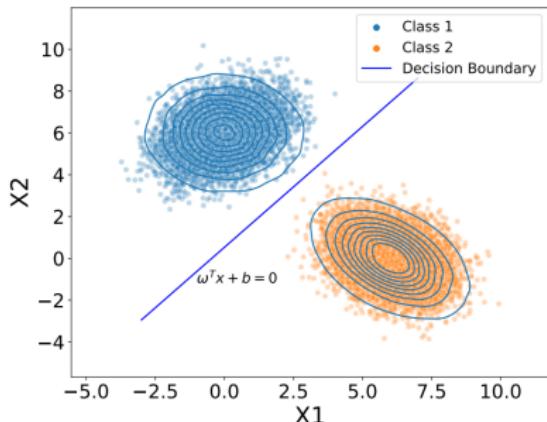


Low-Density Separation Assumption

LDS assumption: classifier makes mistakes in high-density regions.



(a) High-density region



(b) Low-density region.

→ Misclassified samples are closer to decision boundaries.



Logits Reflect Distances to Decision Boundaries

- Decision boundary of class $k \rightarrow \mathcal{H}_k = \{\mathbf{z}' \in \mathbb{R}^q \mid \boldsymbol{\omega}_k^\top \mathbf{z}' = 0\},$



Logits Reflect Distances to Decision Boundaries

- Decision boundary of class $k \rightarrow \mathcal{H}_k = \{\mathbf{z}' \in \mathbb{R}^q \mid \boldsymbol{\omega}_k^\top \mathbf{z}' = 0\},$
- Distance point-hyperplane $d(\mathbf{z}, \mathcal{H}_k) = |\boldsymbol{\omega}_k^\top \mathbf{z}| / \|\boldsymbol{\omega}_k\|,$



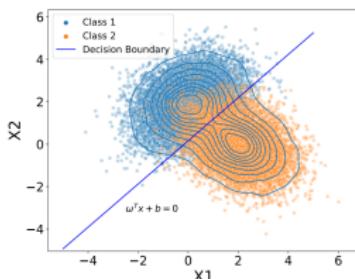
Logits Reflect Distances to Decision Boundaries

- Decision boundary of class $k \rightarrow \mathcal{H}_k = \{\mathbf{z}' \in \mathbb{R}^q \mid \boldsymbol{\omega}_k^\top \mathbf{z}' = 0\},$
- Distance point-hyperplane $d(\mathbf{z}, \mathcal{H}_k) = |\boldsymbol{\omega}_k^\top \mathbf{z}| / \|\boldsymbol{\omega}_k\|,$
- Logits reflect this distance as $|\mathbf{q}_k| = |\boldsymbol{\omega}_k^\top \mathbf{z}| \propto d(\boldsymbol{\omega}_k, \mathbf{z}).$

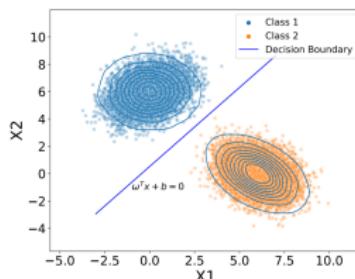


Logits Reflect Distances to Decision Boundaries

- Decision boundary of class $k \rightarrow \mathcal{H}_k = \{\mathbf{z}' \in \mathbb{R}^q \mid \boldsymbol{\omega}_k^\top \mathbf{z}' = 0\}$,
- Distance point-hyperplane $d(\mathbf{z}, \mathcal{H}_k) = |\boldsymbol{\omega}_k^\top \mathbf{z}| / \|\boldsymbol{\omega}_k\|$,
- Logits reflect this distance as $|\mathbf{q}_k| = |\boldsymbol{\omega}_k^\top \mathbf{z}| \propto d(\boldsymbol{\omega}_k, \mathbf{z})$.



(a) High-density region

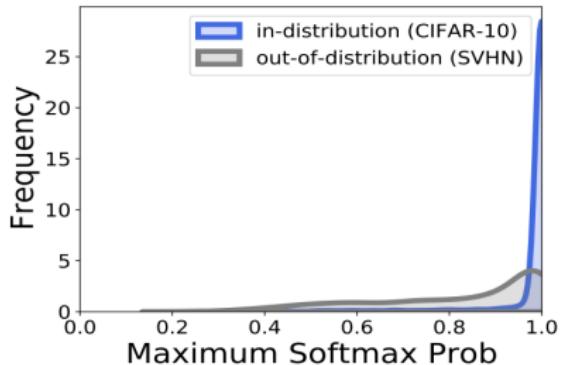


(b) Low-density region.

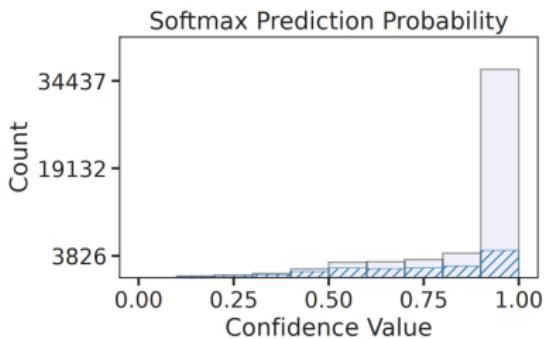
Logits capture the generalization performance.



Softmax Overconfidence



Wei et al.



Odonnat et al.

→ Overconfidence and saturation of softmax outputs.



Prediction Error Accumulation

Logits can be decomposed as follows

$$\underbrace{\mathbf{q}}_{\text{model's logits}} = \underbrace{\mathbf{q}^*}_{\text{ground-truth logits}} + \underbrace{\varepsilon}_{\text{prediction bias}}.$$

Then, the softmax involves computing

$$\exp(\mathbf{q}_{i,k}) = \exp(\mathbf{q}_{i,k}^* + \varepsilon_k) = 1 + (\mathbf{q}_{i,k}^* + \varepsilon_k) + \frac{(\mathbf{q}_{i,k}^* + \varepsilon_k)^2}{2!} + \dots$$



Solution: Truncating the Errors

$$\exp(\mathbf{q}_{i,k}) \approx 1 + (\mathbf{q}_{i,k}^* + \varepsilon_k) + \frac{(\mathbf{q}_{i,k}^* + \varepsilon_k)^2}{2!} + \dots + \frac{(\mathbf{q}_{i,k}^* + \varepsilon_k)^n}{n!}.$$

- ✓ High prediction bias $\varepsilon \rightarrow$ mitigate impact of errors ($n < \infty$)



Solution: Truncating the Errors

$$\exp(\mathbf{q}_{i,k}) \approx 1 + (\mathbf{q}_{i,k}^* + \varepsilon_k) + \frac{(\mathbf{q}_{i,k}^* + \varepsilon_k)^2}{2!} + \dots + \frac{(\mathbf{q}_{i,k}^* + \varepsilon_k)^n}{n!}.$$

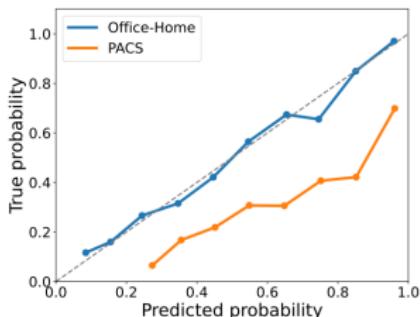
- ✓ High prediction bias $\varepsilon \rightarrow$ mitigate impact of errors ($n < \infty$)
- ✓ Low prediction bias $\varepsilon \rightarrow$ use all the information ($n = \infty$).



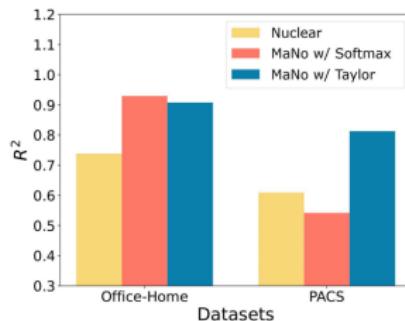
Solution: Truncating the Errors

$$\exp(\mathbf{q}_{i,k}) \approx 1 + (\mathbf{q}_{i,k}^* + \varepsilon_k) + \frac{(\mathbf{q}_{i,k}^* + \varepsilon_k)^2}{2!} + \dots + \frac{(\mathbf{q}_{i,k}^* + \varepsilon_k)^n}{n!}.$$

- ✓ High prediction bias $\varepsilon \rightarrow$ mitigate impact of errors ($n < \infty$)
- ✓ Low prediction bias $\varepsilon \rightarrow$ use all the information ($n = \infty$).



(a) Calibration curves.



(b) Type of normalization.

Trade-off information completeness and error accumulation!



Outline

- ➊ Introduction
- ➋ First Principle Analysis
- ➌ Our Method: MaNo
- ➍ Experimental Results
- ➎ Take Home Message



- ✓ *Input:* Pre-trained model f , test dataset $\mathcal{D}_{\text{test}} = \{\mathbf{x}_i\}_{i=1}^N$.



- ✓ *Input:* Pre-trained model f , test dataset $\mathcal{D}_{\text{test}} = \{\mathbf{x}_i\}_{i=1}^N$.
- ✓ *Inference:* Recover logits $\mathbf{q}_i = f(\mathbf{x}_i)$,



MaNo: A Simple Three-Step Recipe

- ✓ *Input:* Pre-trained model f , test dataset $\mathcal{D}_{\text{test}} = \{\mathbf{x}_i\}_{i=1}^N$.
- ✓ *Inference:* Recover logits $\mathbf{q}_i = f(\mathbf{x}_i)$,
- ✓ *Criterion:* $\Phi(\mathcal{D}_{\text{test}}) = \text{KL}(\text{uniform} \parallel \text{softmax proba})$

$$1) \quad v(\mathbf{q}_i) = \begin{cases} 1 + \mathbf{q}_i + \frac{\mathbf{q}_i^2}{2}, & \text{if } \Phi(\mathcal{D}_{\text{test}}) \leq \eta \\ \exp(\mathbf{q}_i), & \text{if } \Phi(\mathcal{D}_{\text{test}}) > \eta \end{cases}$$

$$2) \quad \sigma(\mathbf{q}_i) = \frac{v(\mathbf{q}_i)}{\sum_{k=1}^K v(\mathbf{q}_i)_k} \in \Delta_K$$

$$3) \quad \mathcal{S}(f, \mathcal{D}_{\text{test}}) = \frac{1}{\sqrt[p]{NK}} \|\mathbf{Q}\|_p = \left(\frac{1}{NK} \sum_{i=1}^N \sum_{k=1}^K |\sigma(\mathbf{q}_i)_k|^p \right)^{\frac{1}{p}}$$



Connection to Uncertainty

Theorem (Xie, O. et al.)

Given a test set $\mathcal{D}_{\text{test}}$ and a pre-trained model f , the estimation score $\mathcal{S}(f, \mathcal{D}_{\text{test}})$ provided by **MaNo** is inversely proportional to the model's uncertainty.

- ✓ Uncertain \rightarrow low accuracy & high entropy \rightarrow low $\mathcal{S}(f, \mathcal{D}_{\text{test}})$,
- ✓ Confident \rightarrow high accuracy & low entropy \rightarrow high $\mathcal{S}(f, \mathcal{D}_{\text{test}})$.

MaNo is positively correlated with the test accuracy.



Outline

- ➊ Introduction
- ➋ First Principle Analysis
- ➌ Our Method: MaNo
- ➍ Experimental Results
- ➎ Take Home Message



SOTA Results & Efficiency

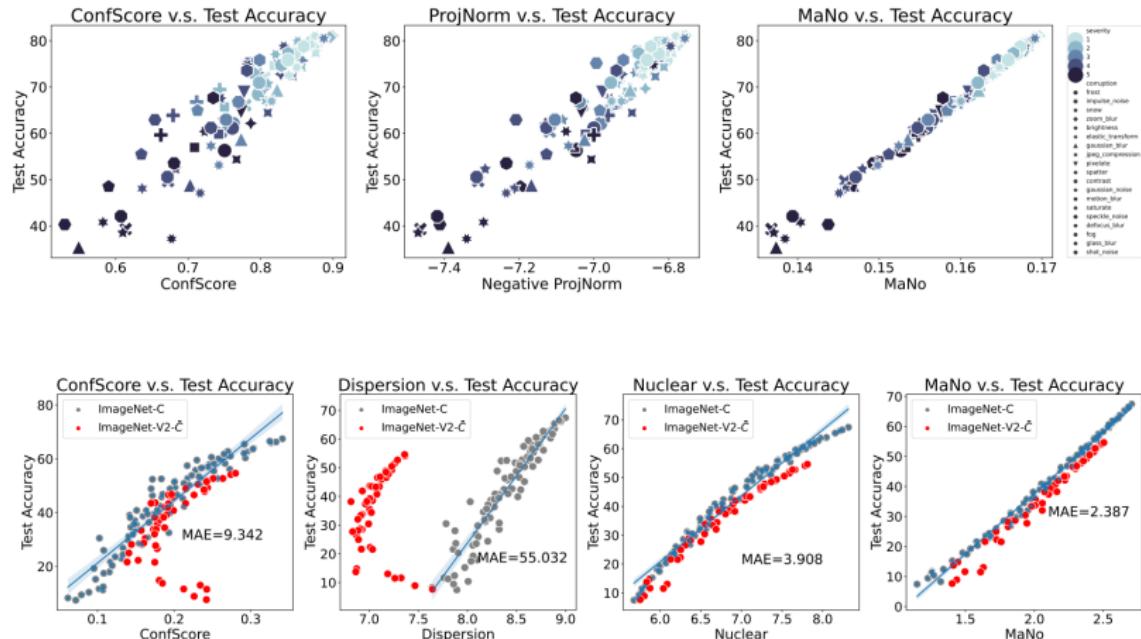
- Comparison with correlation metrics ρ and R^2 ,
- Comparison across architectures: ResNets, ConvNext, ViT,
- Evaluation on common benchmarks and distribution shifts.

Shift	MaNo -	COT 2024	MDE 2024	Nuclear 2023	Dispersion 2023	ProjNorm 2022
Synthetic	0.991	0.988	0.947	0.982	0.960	0.971
Subpopulation	0.983	0.962	0.920	0.973	0.909	0.897
Natural	0.905	0.871	0.436	0.455	0.410	0.382
Overall improvement		2%	25%	6%	26%	28%

MaNo outperforms all the baselines while being training-free.



Qualitative Benefit: Linear Correlation

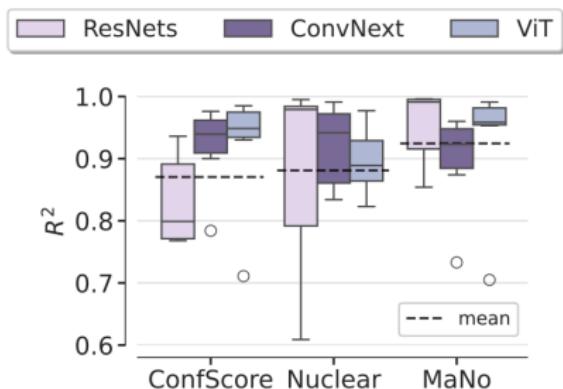
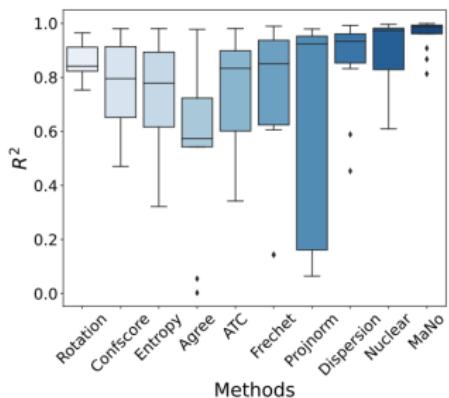


MaNo linearly correlates with the ground-truth performance.



Robustness Analysis

- ✓ Experiments on all distributions shifts,
- ✓ Experiments with various architectures.



MaNo is the best approach to use with SOTA architectures!



Outline

- ➊ Introduction
- ➋ First Principle Analysis
- ➌ Our Method: MaNo
- ➍ Experimental Results
- ➎ Take Home Message



Take Home Message

- ✓ Predicting accuracy under distribution shifts is challenging.



Take Home Message

- ✓ Predicting accuracy under distribution shifts is challenging.
- ✓ Most methods use logits and fail under miscalibration.



Take Home Message

- ✓ Predicting accuracy under distribution shifts is challenging.
- ✓ Most methods use logits and fail under miscalibration.
- ✓ **MaNo** → theoretically grounded estimation approach.



Take Home Message

- ✓ Predicting accuracy under distribution shifts is challenging.
- ✓ Most methods use logits and fail under miscalibration.
- ✓ **MaNo** → theoretically grounded estimation approach.
- ✓ Benefits: **SOTA, efficient, architecture agnostic, robust.**



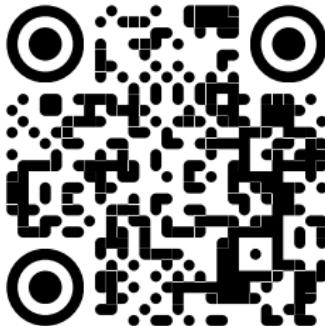
To Know More

This work has been accepted at NeurIPS 2024.

Paper: <https://arxiv.org/pdf/2405.18979>

Code: <https://github.com/Renchunzi-Xie/MaNo>

To know more about my research, check out my website!



ambroiseodt.github.io

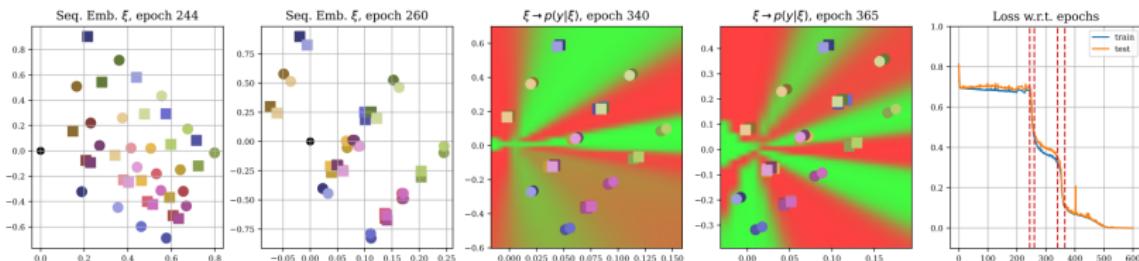


Clustering Head: A Visual Case Study of the Training Dynamics in Transformers



Self-Promotion

Using our visual sandbox, we identify **clustering heads**, circuits that learn the invariance of the sparse addition modular task and study their training dynamics.



Paper: <https://arxiv.org/pdf/2410.24050>

Code: <https://github.com/facebookresearch/pal>

Thank you for your attention!