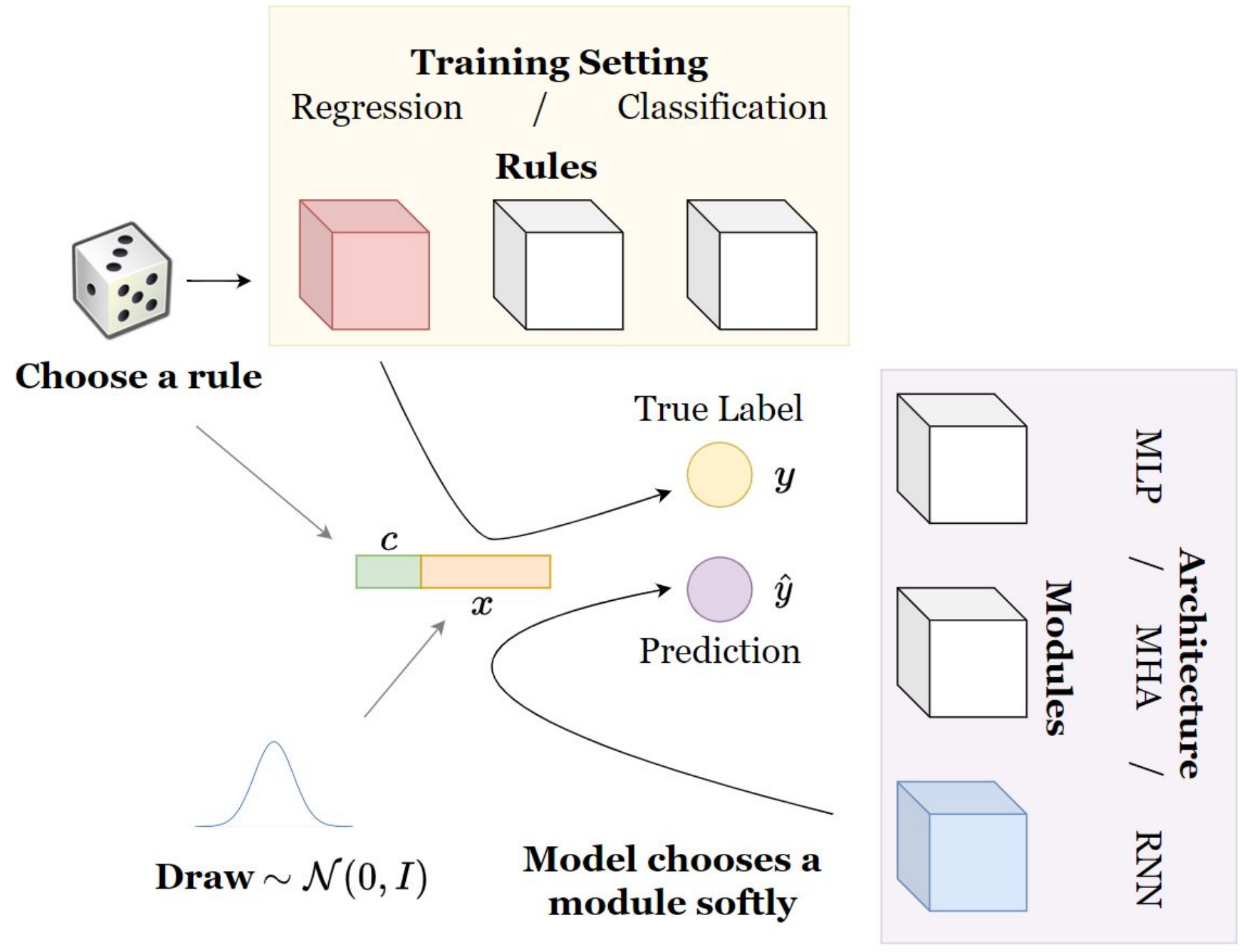
Is a Modular Architecture Enough?

Sarthak Mittal, Yoshua Bengio, Guillaume Lajoie

Motivation

A key intuition behind the success of modular systems is that the data generating system for most real-world settings consists of sparsely interacting parts. However, the field has been lacking in a rigorous quantitative assessment of such systems since the real-world data distributions are complex and unknown.



We provide a thorough assessment of common modular architectures through the lens of simple modular data distributions. We highlight the benefits of modularity and sparsity and reveal insights on the challenges faced while optimizing them. In doing so, we propose evaluation metrics that highlight the regimes in which the benefits of modular systems are substantial, as well as the sub-optimality of current end-to-end learned modular systems as opposed to their claimed potential.

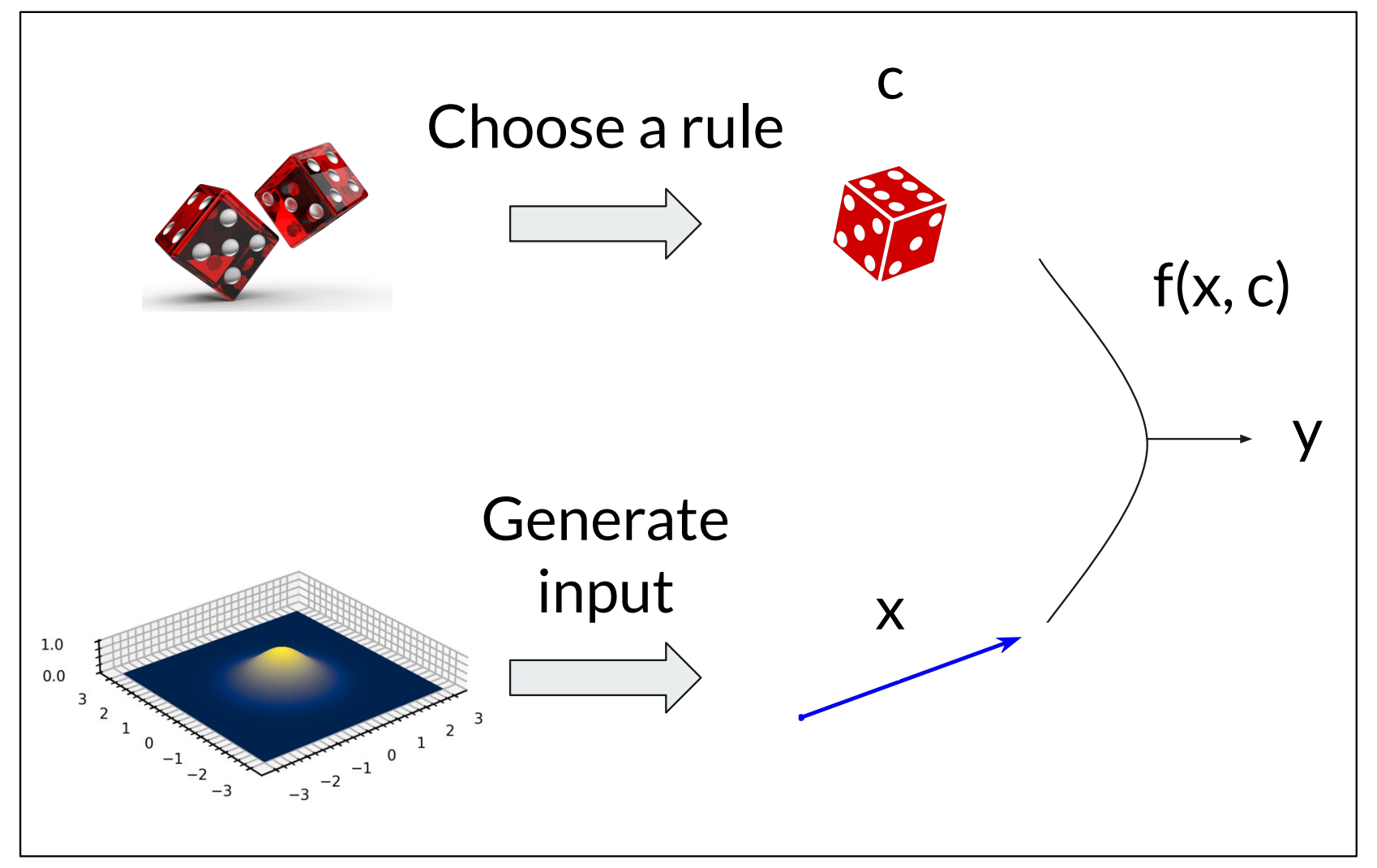
Contributions

- Develop tasks and metrics to quantify collapse and specialization.
- Distill modular inductive biases and systematically evaluate them.
- Highlight the benefits of specialization, especially in the regime of many rules.
- Uncover the sub-optimality of standard modular architectures.

Data-Generating Process

We consider known synthetic mixture of experts based data for different kinds of data symmetries. eg.

- Binary operations (*for MLPs*)
- Set operations (*for MHAs*)
- Time Series operations (*for RNNs*)



We use different data-generating processes according to the base template described above for regression as well as classification settings. Eg. for binary operations it can be mixture of different linear combinations while for time series operations it can be a switching linear dynamical system.

Models

We consider four different types of models which can be characterized as

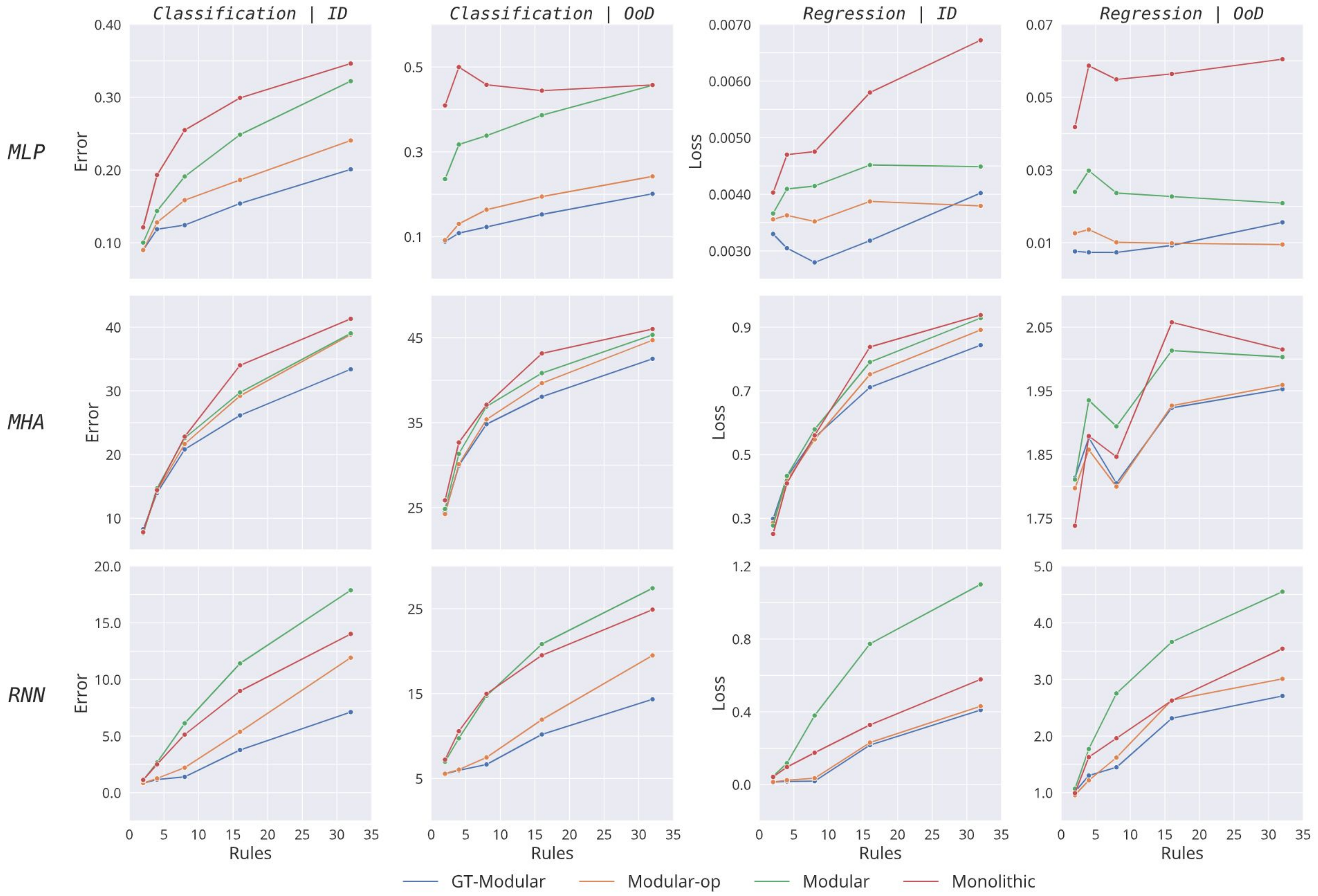
- *Monolithic:* A single big model.
- *Modular:* A mixture-of-experts based system without any privileged information.
- *Modular-op:* Modular system with specialization driven solely by privileged rule information.
- *GT-Modular:* Modular system with oracle, baked-in perfect specialization.

Model	Functional Form
Monolithic	$\hat{\mathbf{y}} = f(\mathbf{x}, \mathbf{c})$
Modular	$\hat{\mathbf{y}}_m, p_m = f_m(\mathbf{x}, \mathbf{c})$ $\hat{\mathbf{y}} = \sum_{m=1}^{R} p_m \, \hat{\mathbf{y}}_m$
Modular-op	$\hat{\mathbf{y}}_m = f_m(\mathbf{x}, \mathbf{c})$ $\mathbf{p} = g(\mathbf{c})$ $\hat{\mathbf{y}} = \sum_{m=1}^{R} p_m \, \hat{\mathbf{y}}_m$
GT-Modular	$\hat{\mathbf{y}}_m = f_m(\mathbf{x}, \mathbf{c})$ $\hat{\mathbf{y}} = \sum_{m=1}^{R} c_m \, \hat{\mathbf{y}}_m$

For fair comparison, we control for the number of parameters between different models

Performance

In-Distribution and Out-of-Distribution Performance



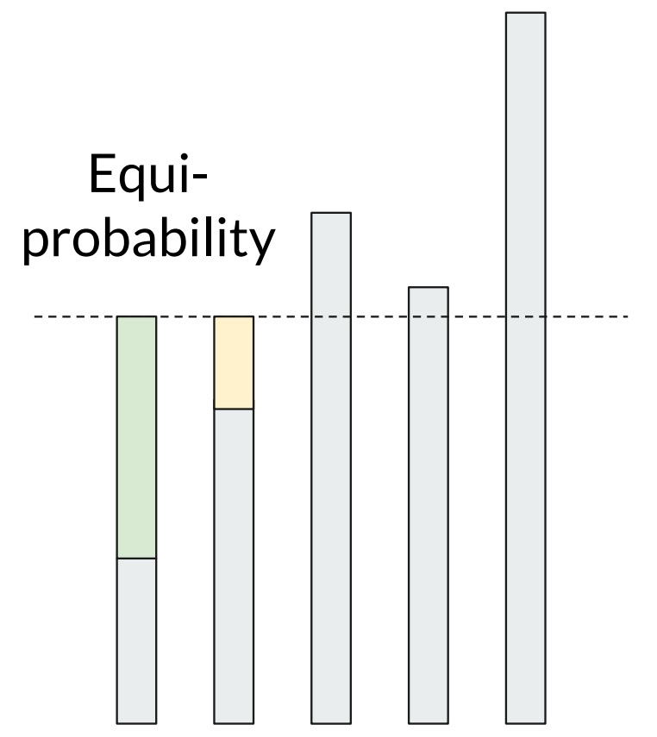
We see that oracle specialization leads to significant improvements, especially when the number of rules are large. However, typical modular systems without privileged information are not able to reach those benefits.

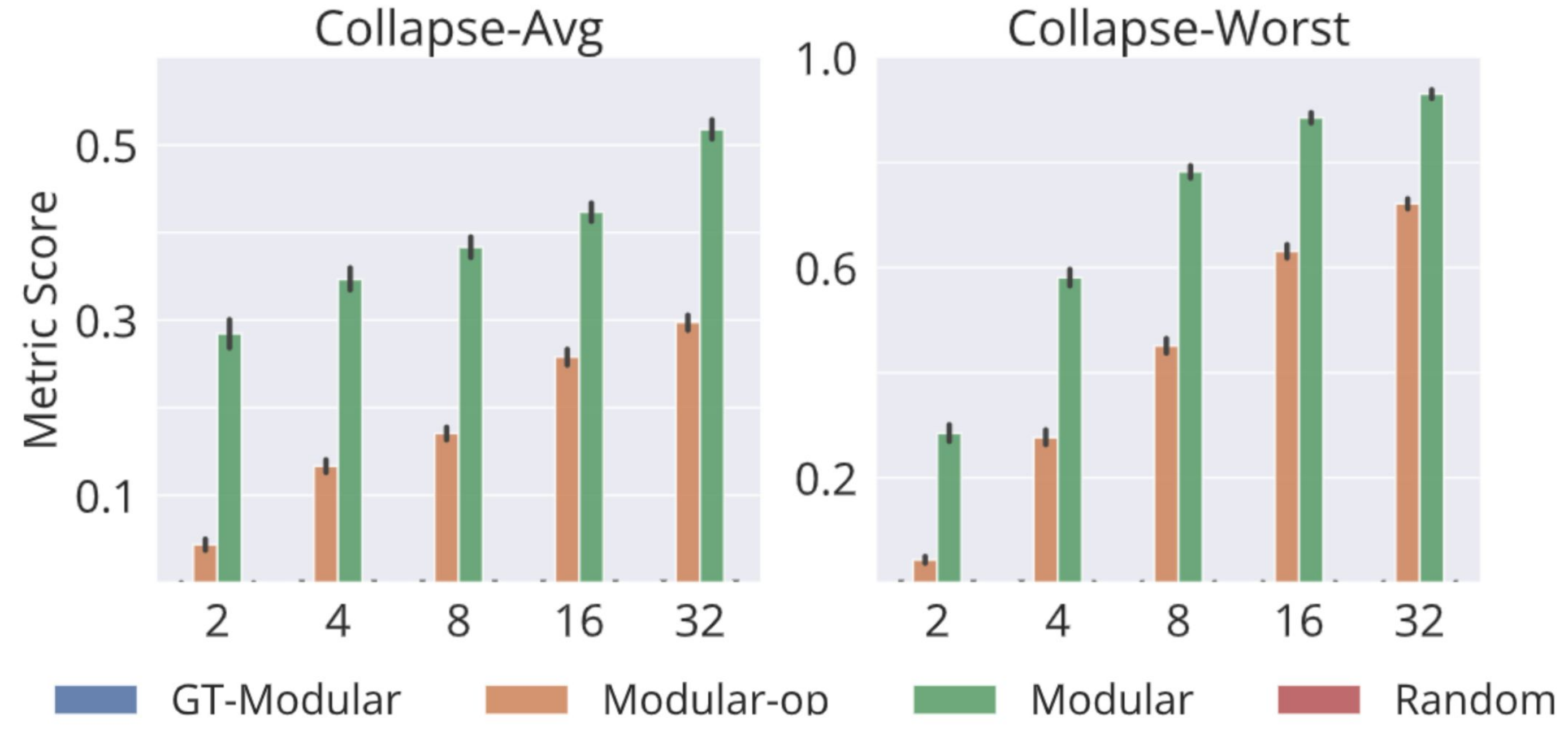
Collapse

Measure the amount of under-utilization of modules

Collapse-Worst: The worst case collapse that modules see; i.e. solely based on the least used module.

Collapse-Average: The average case collapse which considers all modules and computes their general under-utilization.





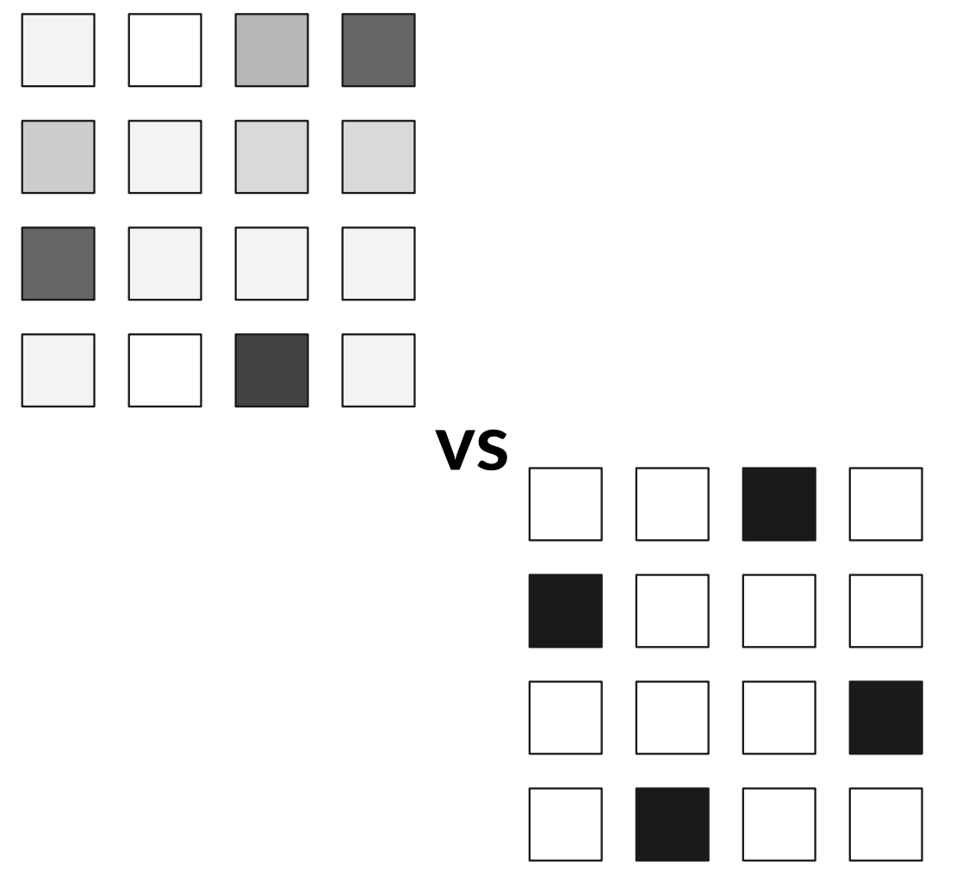
Specialization

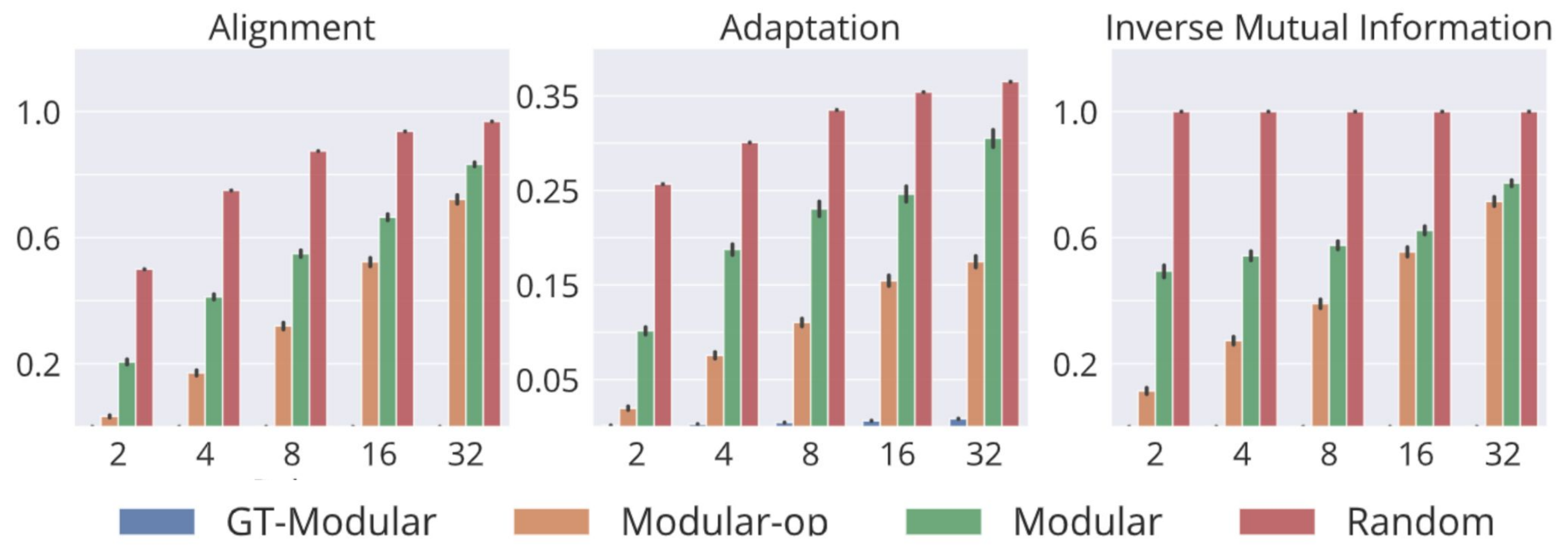
Measure the alignment of modules to oracle rules

Alignment: Computes distance with the closest permutation matrix.

Inverse Mutual Information: Computes the mutual information between the modules and rules.

Adaptation: Computes the extent of flexibility in module activations when the rule distribution is altered.




Alignment
Adaptation
Inverse Mutual Information
GT-Modular
Modular-op
Modular
Random