THE GRADIENT-CAUSAL GAP: WHY GRADIENT IMPORTANCE FAILS ON COMPLEX TASKS

Donald Ye
Independent Researcher
donaldye827@gmail.com

ABSTRACT

Removing ”important” high-gradient components from a neural network can improve generalization, while removing unimportant” low-gradient components can destroy it. We demonstrate this paradox by formalizing the *Gradient-Causal Gap* in Transformers trained on algorithmic tasks. While gradient magnitude and causal importance align on simple tasks ($\rho = 0.73$ for reversal), this relationship collapses as task complexity increases ($\rho = 0.32$ for sorting), sometimes becoming inverted ($\rho = -0.11$). Pruning experiments reveal that gradient magnitude is not merely inaccurate but *unpredictably* so. Removing low-gradient ”Hidden Heroes” consistently devastates OOD accuracy (-32%). Removing high-gradient ”Gradient Bloats” is a coin flip: harmless in most seeds (indicating optimization noise), catastrophic in others (indicating overfitting circuits). This unpredictability means gradient-based pruning cannot reliably preserve model capabilities. [1]

1 INTRODUCTION

Removing “important” components from a neural network should hurt performance. Yet we find the opposite. Pruning high-gradient components can *improve* out-of-distribution (OOD) generalization, while removing low-gradient components can destroy it. This challenges a core assumption in interpretability which is that gradient magnitude indicates component importance.

Gradient-based importance underpins pruning methods (Han et al., 2015; Molchanov et al., 2017) and attribution techniques (Sundararajan et al., 2017), with the underlying logic that large gradients indicate large contributions to reducing training loss. But importance *for what*? Gradients optimize for training loss, not generalization.

We investigate this by comparing gradient-based importance to causal importance (measured via ablation) in Transformers trained on algorithmic tasks. We treat each attention head and MLP sublayer as a distinct component, measuring gradient magnitude and causal effect independently for each. Using these tasks as a controlled testbed for systematic generalization, we formalize the **Gradient-Causal Gap**. This gap is the divergence between what gradients signal as important and what causally matters for OOD generalization.

This gap produces two failure modes:

- **Gradient Bloats**: components with high gradient but low causal impact
- **Hidden Heroes**: components with low gradient but high causal impact

Our experiments show:

- Gradient-causal correlation drops from $\rho = 0.73$ (reversal) to $\rho = 0.32$ (sorting), sometimes becoming negative ($\rho = -0.11$).
- Hidden Heroes cluster in later layers (Layer 3); Gradient Bloats cluster in early layers (Layers 0–1).
- Pruning Hidden Heroes drops OOD accuracy by 32%. Pruning Gradient Bloats improves it by 3.5%.

[1] Code: https://anonymous.4open.science/r/ICLR_2026_LIT-workshop_CG-D42B

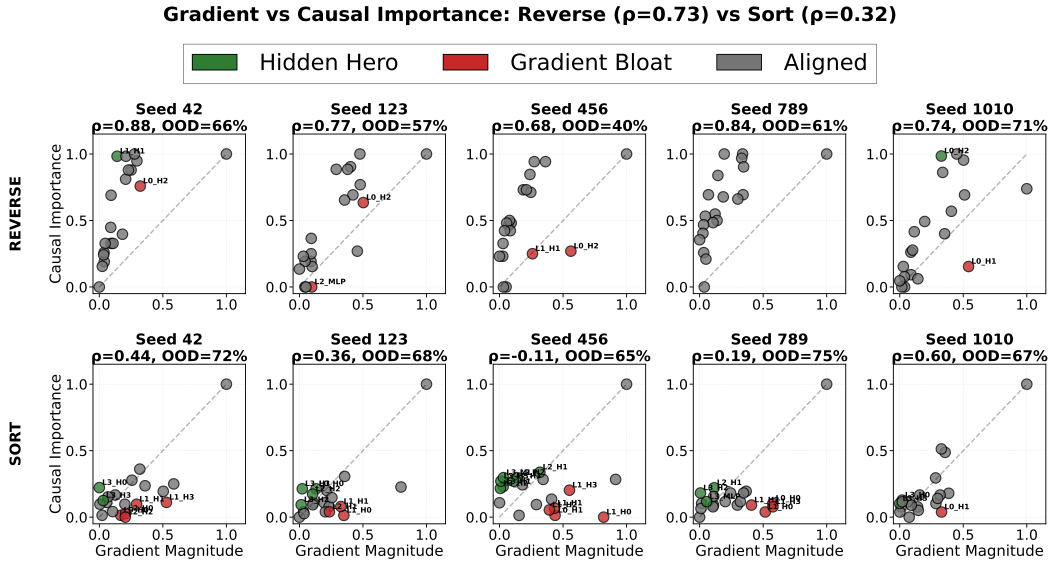

Figure 1: **The Gradient-Causal Gap**. Gradient magnitude aligns with causal importance in Reverse ($\rho = 0.73$), but collapses in Sort ($\rho = 0.32$). Scores are min-max normalized per seed; coordinates (1.0,y) and (x,1.0) denote the maximum gradient and causal components. See Figure 4 in Appendix A.2.

In our setting, gradients track training dynamics rather than the circuits needed for generalization. For mechanistic interpretability, understanding implicit reasoning without chain-of-thought (Pfau et al., 2024), suggests standard importance measures may miss the components that matter most, with implications for both understanding and compressing models that generalize.

2 Method

2.1 Task and Model Architecture

We evaluate the *Gradient-Causal Gap* using two algorithmic benchmarks: **Sequence Reversal** and **Sequence Sorting**. Both tasks use an alphabet of integers $x \in \{1, \dots, 99\}$. We use a decoder-only Transformer with $L = 4$ layers and $H = 4$ heads per layer. Detailed hyperparameters and training configurations are provided in Appendix A.

2.2 Measuring Importance

We define the set of model components $\mathcal{S} = \{H_{\ell,h}\} \cup \{M_\ell\}$, where $H_{\ell,h}$ denotes the h-th attention head in layer ℓ, and M_ℓ denotes the MLP sublayer in layer ℓ. For our architecture, $|\mathcal{S}| = 20$.

To isolate components for generalization, we evaluate models on out-of-distribution (OOD) lengths $N \in \{8, 9, 10, 11\}$. For each seed, we select the specific length where the model achieves accuracy within $[20\%, 75\%]$, ensuring the model logic is active but prone to measurable failures. This range ensures the model has learned partial structure but remains susceptible to failures. Accuracy is measured as exact sequence match.

Gradient Magnitude (G). For each component $i \in \mathcal{S}$, we compute the average Frobenius norm of the gradients with respect to the component's weights $\mathbf{W}_i$ across 50 OOD batches:

$$G_i = \frac{1}{B} \sum_{b=1}^{B} \|\nabla_{\mathbf{W}_i} \mathcal{L}_b\|_F \tag{1}$$

where $B = 50$ and $\mathcal{L}_b$ is the loss for batch b. For attention heads, $\mathbf{W}_i$ corresponds to the Value weights $\mathbf{W}_V \in \mathbb{R}^{d_{\text{model}} \times d_{\text{head}}}$; for MLPs, $\mathbf{W}_i$ refers to the weights of the second linear layer. While gradient magnitude technically measures sensitivity to parameter changes rather than contribution to output, it remains a dominant heuristic for pruning (Han et al., 2015) We test whether this heuristic holds for OOD generalization.

Causal Importance (C). We measure the functional necessity of a component via *Mean Ablation*. We compute the mean activation $\boldsymbol{\mu}_i$ of component i across the OOD distribution. During inference, we replace the component's output $\mathbf{x}_i$ with $\boldsymbol{\mu}_i$ and measure the resulting accuracy drop:

$$C_i = \text{Acc}_{\text{base}} - \text{Acc}_{\text{ablated}(i \to \boldsymbol{\mu}_i)} \tag{2}$$

where Acc_{base} represents the exact-match sequence accuracy on the OOD length prior to ablation.

2.3 The Gradient-Causal Gap and Classification

We quantify the misalignment between optimization signals and functional importance by computing the Spearman correlation ρ between $\mathbf{G}$ and $\mathbf{C}$. We define the **Gradient-Causal Gap** (Δ_i) as the difference in ordinal ascending ranks:

$$\Delta_i = \text{Rank}(G_i) - \text{Rank}(C_i) \tag{3}$$

Under this definition, a large negative value indicates that a component is ranked significantly higher in causal importance than its gradient magnitude would suggest. Components are classified into three categories based on Δ_i:

- **Hidden Heroes ($\Delta_i \leq -6$):** Low-gradient components that are causally essential.
- **Gradient Bloats ($\Delta_i \geq 6$):** High-gradient components with negligible causal impact.
- **Aligned ($|\Delta_i| < 6$):** Components where gradient magnitude predicts causal importance.

To validate the roles of these categories, we perform pruning interventions by ablating the top two components from the Hero and Bloat categories and observing the impact on OOD generalization.

3 Results

3.1 The Gradient-Causal Gap

Gradient-causal alignment degrades as task complexity increases. On Reversal, gradient magnitude correlates strongly with causal importance ($\rho = 0.726 \pm 0.121$), with correlations positive across all 10 seeds. On Sorting, this relationship weakens ($\rho = 0.318 \pm 0.241$), with higher variance and one seed exhibiting negative correlation ($\rho = -0.113$).

The number of misaligned components also increases with complexity. Reversal produces fewer misaligned instances (7 Hidden Hero and 17 Gradient Bloat occurrences across 10 seeds), while Sorting produces nearly five times as many Hidden Hero occurrences (33 total) and twice as many Gradient Bloat occurrences (36 total).

Table 1: Summary of gradient-causal alignment across tasks. Values show mean $\pm$ std over 10 seeds.

Task	Spearman ρ	ρ Range	Hero Count	Bloat Count
Reversal	0.726 ± 0.121	$[0.47, 0.88]$	7	17
Sorting	0.318 ± 0.241	$[-0.11, 0.73]$	33	36

3.2 Layer-wise Patterns

Hidden Heroes and Gradient Bloats cluster in distinct layers. In Sorting, Hidden Heroes concentrate in later layers (L2–L3), appearing most frequently in Layer 3 heads. Gradient Bloats cluster in early layers (L0–L1), with Layer 1 heads dominating (Figure 2). Full component breakdowns are provided in Appendix A.2.

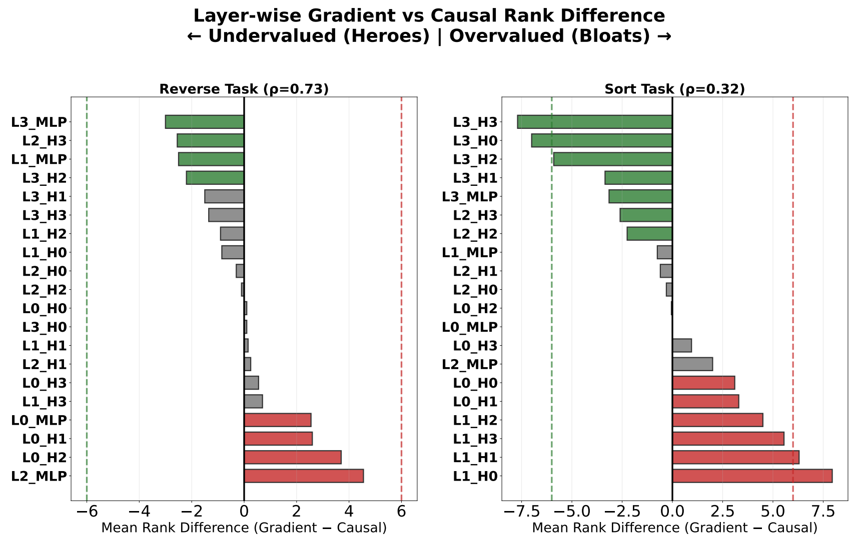

Figure 2: Layer-wise distribution of Hidden Heroes and Gradient Bloats in Sorting. Heroes cluster in later layers (L2–L3), while Bloats concentrate in early layers (L0–L1).

This pattern suggests a division of laborm where early layers perform high-gradient operations that contribute to training loss reduction but are not critical for OOD generalization, while later layers implement the core algorithmic logic with lower gradient signal.

3.3 Pruning Consequences

Targeted pruning confirms the functional nature of these components. Pruning Hidden Heroes consistently devastates generalization (OOD accuracy drops by 32%).

However, the role of Gradient Bloats reveals a deeper misalignment. Across 10 seeds, pruning these high-gradient components did not produce a uniform effect, but rather split into two distinct functional regimes. In the majority of runs, pruning Gradient Bloats had negligible impact on in-distribution (ID) accuracy ($\Delta > -5\%$), identifying them as **Optimization Noise**, artifacts of the training process with no functional role. Conversely, in the remaining seeds, pruning caused significant ID collapse (up to 39% drop), identifying them as **Overfitting Circuits**. This bimodality confirms that a high gradient norm is an ambiguous signal and may flag as shortcut, with no way to distinguish them. Full seed-by-seed ID/OOD effects are reported in Appendix A.

4 Discussion

4.1 Why the Gap Emerges

The gap arises because gradients track optimization dynamics, not functional logic. High-gradient components often act as broad feature extractors, accumulating backpropagation signals from all token positions (Molchanov et al., 2019). In contrast, "Hidden Heroes" implement sparse, localized logic that yields weak gradients. This is not a depth artifact as the Reversal task exhibits late-layer Bloats (L2_MLP) and early-layer Heroes (L1_MLP), confirming the gap is driven by task semantics.

The Hazard of Gradient Ambiguity. The bimodal nature of Gradient Bloats poses a fundamental risk for model compression and interpretability. Because high-gradient components fluctuate stochastically between being shortcuts and noise, gradient-based pruning offers no guarantees. This is where the same method that harmlessly compresses one model (Seed 42) destroys another (Seed 789). This unpredictability suggests that gradient magnitude is insufficient for identifying safe-to-prune components without causal verification.

4.2 Implications for Interpretability

The observed divergence poses a challenge to the reliability of gradient-based interpretability. Methods like Integrated Gradients (Sundararajan et al., 2017) assume that gradient magnitude reflects

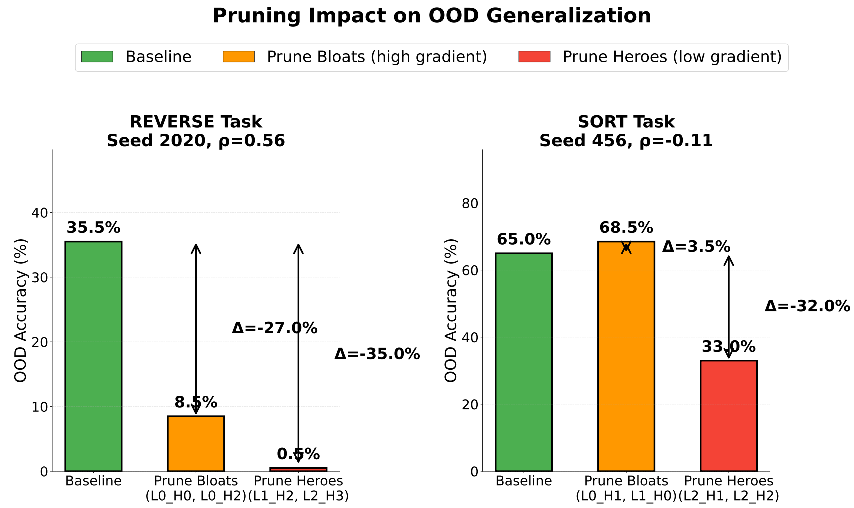

Figure 3: Pruning effects on OOD accuracy for Sorting (Seed 456). Removing low-gradient "Hidden Heroes" devastates generalization. In this seed, pruning high-gradient "Gradient Bloats" paradoxically improves OOD accuracy ($\Delta = +3.5\%$), illustrating the *Optimization Noise* regime where gradients highlight functionally redundant components.

contribution to model output, but our results show this assumption breaks down for OOD generalization. If we use gradient saliency to identify important components, we map training dynamics rather than reasoning circuits. Our findings reinforce the need for causal interventions. Gradient-based methods are efficient but cannot replace the ground-truth signal from ablations or activation patching (Vig et al., 2020), especially for models intended for robust deployment.

4.3 Connection to Implicit Reasoning

When models solve tasks without Chain-of-Thought, reasoning is implemented in component activations. If Hidden Heroes drive this implicit logic, then current compression techniques may prune the components that enable generalization. This finding carries direct implications for the LIT workshop theme. The circuits implementing implicit reasoning (Wang et al., 2023) may be precisely those that standard importance measures undervalue.

4.4 Limitations

We use small Transformers (4 layers, 4 heads) on algorithmic tasks. Whether the Gradient-Causal Gap persists at scale remains open. We also use mean ablation as our causal measure; activation patching may yield different rankings. Our tasks have clear algorithmic structure—the gap may behave differently on naturalistic tasks.

5 Conclusion

We formalize the Gradient-Causal Gap, the divergence between what gradients signal as important and what causally matters for OOD generalization. On algorithmic tasks, this gap widens with complexity, producing Hidden Heroes and Gradient Bloats in distinct layers. Pruning confirms that removing Hidden Heroes devastates generalization while removing Gradient Bloats can improve it. Importance is not a single scalar, distinguishing components that drive training from those that drive reasoning remains a critical problem for interpretability and compression.

Acknowledgments

We used Claude for assistance with code development and debugging. We used Gemini for language editing and refinement of author-written content. All experimental design, methodology, analysis, and interpretation are the authors' own work.

REFERENCES

Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. *NeurIPS*, 2015.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *ICLR*, 2017.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11264–11272, 2019.

Jacob Pfau, William Merrill, and Samuel R Bowman. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758*, 2024.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Naomi Saphra, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401, 2020.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*, 2023.

A SUPPLEMENTARY MATERIALS

A.1 IMPLEMENTATION DETAILS AND HYPERPARAMETERS

This section details the architectural choices and training regimen used to produce the models analyzed in the main text. We emphasize the use of a wide range of random seeds to ensure the Gradient-Causal Gap is a structural property of the optimization landscape rather than an initialization artifact.

Hyperparameters. The Transformer uses $d_{\text{model}} = 128$, $d_{\text{ff}} = 512$, and learned positional embeddings. We use the Adam optimizer with a learning rate of 10^{-3} and a batch size of 64.

Training. Training sequences have lengths $N \in [3, 7]$. Models are trained until they reach a target accuracy of $\geq 90\%$ on the training distribution or hit a maximum of 15,000 steps. We utilize a cross-entropy loss function, ignoring padding tokens during calculation. Experiments are conducted over 10 random seeds (Seeds: 42, 123, 456, 789, 1010, 2020, 3030, 4040, 5050, 6060) to ensure the stability of the observed Gradient-Causal Gap.

A.2 FULL SEED-BY-SEED EXPERIMENTAL LOGS

Figure 4 illustrates the individual Spearman ρ values for all 20 experimental runs. While Reversal maintains a tight, high-correlation cluster, Sorting exhibits significant variance, including the negative correlation observed in Seed 456.

A.3 PRUNING RESULTS

Table 2: Pruning effects on OOD accuracy. We ablate the top-2 components from each category.

Task (Seed)	**Baseline**	**Prune Heroes**	**Prune Bloats**
Sorting (456)	65.0%	33.0% (↓32.0)	68.5% (↑3.5)
Reversal (2020)	35.5%	0.5% (↓35.0)	8.5% (↓27.0)

In Sorting (Seed 456), pruning Gradient Bloats improves OOD accuracy, confirming these components are not only unimportant but potentially harmful to generalization. In Reversal, where gradient-causal alignment is stronger, both pruning interventions hurt performance—Gradient Bloats in this regime still carry partial causal relevance.

Table 3: Most frequent Hidden Heroes and Gradient Bloats across 10 seeds.

Task	**Component**	**Frequency**
Hidden Heroes		
Sorting	L3_H3	7/10
Sorting	L3_H0	5/10
Sorting	L3_H2	5/10
Sorting	L2_H1	3/10
Sorting	L2_H2	3/10
Gradient Bloats		
Sorting	L1_H0	6/10
Sorting	L1_H1	6/10
Sorting	L1_H2	6/10
Sorting	L1_H3	5/10
Sorting	L0_H0	3/10

Table 4: Seed-by-seed results for Sorting task.

Seed	**OOD Len**	**OOD Acc**	ρ	**Heroes / Bloats**
42	9	72%	0.438	2 / 5
123	9	68%	0.364	4 / 3
456	9	65%	-0.113	7 / 5
789	9	75%	0.191	3 / 4
1010	9	67%	0.600	2 / 1
2020	8	73%	0.098	3 / 4
3030	8	68%	0.726	2 / 1
4040	8	64%	0.368	4 / 3
5050	10	59%	0.082	3 / 6
6060	9	69%	0.423	3 / 4

Table 5: Seed-by-seed results for Reversal task.

Seed	**OOD Len**	**OOD Acc**	ρ	**Heroes / Bloats**
42	9	66%	0.883	1 / 1
123	9	57%	0.767	0 / 2
456	9	40%	0.678	0 / 2
789	9	61%	0.844	0 / 0
1010	9	71%	0.737	1 / 1
2020	9	45%	0.559	2 / 4
3030	9	52%	0.470	0 / 3
4040	9	35%	0.793	1 / 2
5050	8	64%	0.726	2 / 1
6060	9	47%	0.805	0 / 1

Table 6: Effect of pruning Gradient Bloats on in-distribution (ID) accuracy for Sorting. Negative drop indicates pruning *improved* performance. Seed 3030 excluded due to no valid OOD length. Mean ID Drop: $2.9\% \pm 14.4\%$.

Seed	# Bloats	ID Base	ID Pruned	Drop
42	4	90.0%	96.0%	−6.0%
123	2	94.0%	98.0%	−4.0%
456	3	92.0%	86.0%	+6.0%
789	3	91.0%	52.0%	+39.0%
1010	3	93.0%	98.0%	−5.0%
2020	3	87.0%	96.0%	−9.0%
4040	3	97.0%	94.0%	+3.0%
5050	3	90.0%	86.0%	+4.0%
6060	3	94.0%	96.0%	−2.0%
Mean Drop				$2.9\% \pm 14.4\%$

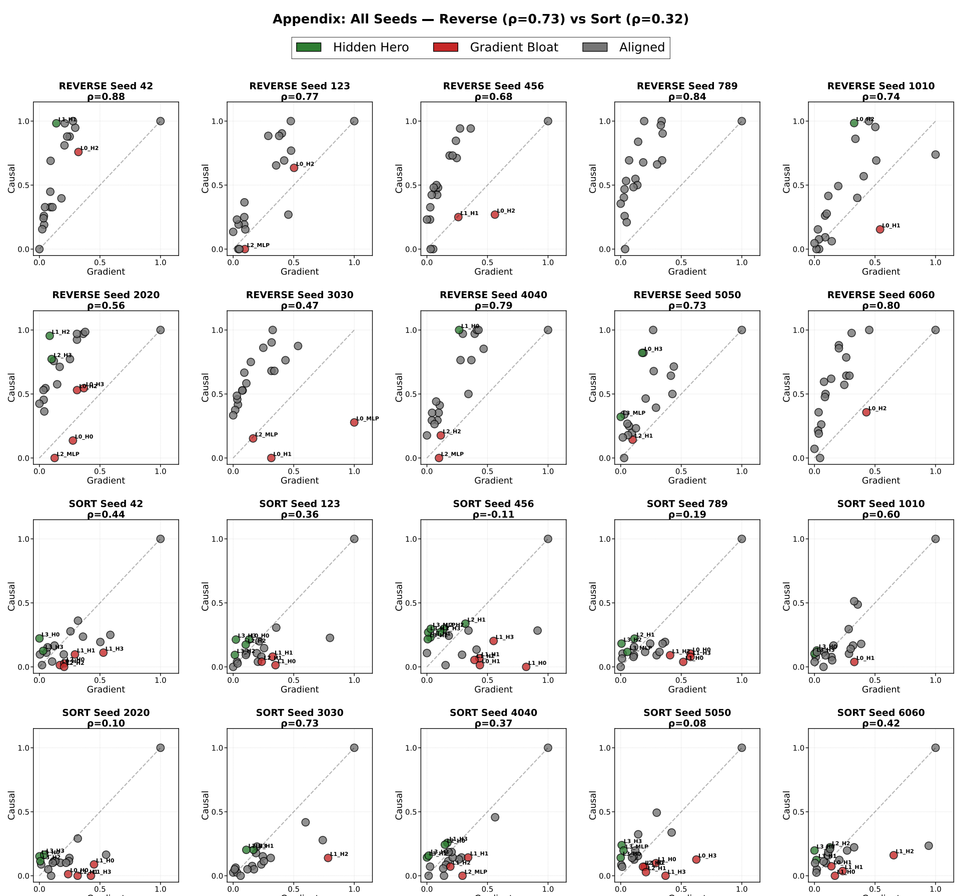

Figure 4: Spearman ρ for 10 seeds of Reversal vs. 10 seeds of Sorting. The divergence confirms that as complexity increases, gradient magnitude becomes a stochastic rather than deterministic proxy for causal importance.