HEDONIC NEURONS: A MECHANISTIC MAPPING OF LATENT COALITIONS IN TRANSFORMER MLPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuned Large Language Models (LLMs) encode rich task-specific features, but the form of these representations—especially within MLP layers—remains unclear. Empirical inspection of LoRA updates shows that new features concentrate in mid-layer MLPs, yet the scale of these layers obscures meaningful structure. Prior probing suggests that statistical priors may strengthen, split, or vanish across depth, motivating the need to study how neurons *work together* rather than in isolation.

We introduce a mechanistic interpretability framework based on *coalitional game theory*, where neurons mimic agents in a hedonic game whose preferences capture their synergistic contributions to layer-local computations. Using top-responsive utilities and the PAC-Top-Cover algorithm, we extract *stable coalitions of neurons*—groups whose joint ablation has non-additive effects—and track their transitions across layers as persistence, splitting, merging, or disappearance.

Applied to LLaMA, Mistral, and Pythia rerankers fine-tuned on scalar IR tasks, our method finds coalitions with consistently higher synergy than clustering baselines. By revealing how neurons cooperate to encode features, hedonic coalitions uncover higher-order structure beyond disentanglement and yield computational units that are functionally important, interpretable, and predictive across domains.

1 INTRODUCTION

Consider a large language model fine-tuned to compute semantic similarity between text pairs. When presented with two sequences, the model outputs a scalar score, e.g. 0.76. But how is this number internally computed? Within the millions of parameters of a transformer, such decisions are not the work of isolated neurons, but of groups that cooperate to represent abstract features like "semantic overlap," "term frequency patterns," or "syntactic alignment." These computations may parallel familiar retrieval metrics – e.g., some neuron coalitions might compute TF-IDF-like statistics (Sparck Jones, 1972), others might capture cosine similarities between representations, while still others might encode position-dependent matching signals. Traditional interpretability methods have limitations here: probing (Gurnee et al., 2023; Hewitt & Manning, 2019) captures correlations with labels but ignores cooperation, sparse autoencoders (SAEs) (Huben et al., 2024) disentangle activations into monosemantic directions but overlook nonlinear dependencies, and clustering (Cao et al., 2025; Song et al., 2024) groups neurons by statistical proximity rather than functional interaction. What is missing is a principled way to identify *synergistic neuron groups*—subsets whose combined contribution exceeds the sum of their parts. We aim to identify the computational units that organize into stable coalitions, that potentially encode these mathematical concepts within scalar-output LLMs.

Recent work has shown that LoRA fine-tuning can teach LLMs new tasks by updating only mid-level MLP layers, nearly matching full fine-tuning (Hu et al., 2022; Zhou et al., 2024; Nijasure et al., 2025). Yet inspection of these LoRA weight updates reveals little obvious structure: millions of parameters diffuse across neurons, obscuring which units encode task-specific features. We hypothesize that the key to isolating LoRA emergent behaviour lies in identifying *coalitions* of neurons that consistently co-adapt under fine-tuning. Inspired by game theory, we model neurons as agents in a *hedonic game* (Dreze & Greenberg, 1980), where preferences reflect synergy with others. Though neurons are not literally rational, stochastic gradient descent imposes a form of selection pressure: directions that reduce loss persist, and many neurons are only useful in combination (e.g., a feature computation). Thus, stable coalitions naturally emerge as groups of neurons that survive training together. This

evolutionary analogy motivates the hedonic game framing: utilities capture how much a neuron's survival depends on its synergy with others, and stable coalitions correspond to groups of neurons that consistently co-adapt under training. By modeling these groups as coalitions, we open a path toward reverse-engineering their function and symbolically characterizing their emergent behavior.

Why does this matter? Beyond offering a new perspective on interpretability, coalition analysis provides actionable insight into how task-specific features are represented and evolve. By showing which neuron groups are functionally indispensable, our framework suggests future directions for practical interventions such as model comparison, transfer learning, or modular editing at the coalition level rather than at the level of individual weights. Moreover, tracking persistence, splits, and vanishings highlights how statistical priors are refined or discarded across depth, shedding light on the internal dynamics of fine-tuned models—information that clustering or SAE-style methods cannot reveal. Thus, stable coalitions are not only theoretically appealing but also open a path to understanding and eventually controlling the computational units that fine-tuning creates.

Our Contributions. We introduce a game-theoretic framework for discovering and analyzing neuron coalitions in transformer MLPs. (1) We model neurons as players in a hedonic cooperative game with additively separable utilities based on synergy, and solve for ε-PAC-stable outcomes using the PAC-Top-Cover algorithm. (2) We evaluate coalitions both intrinsically and extrinsically: compared to clustering baselines, they achieve +0.29 *Pairwise* and +0.49 *Ratio* synergy, exhibit 3–5$\times$ larger out-of-distribution performance drops under ablation, align more strongly with IR heuristics (BM25, IDF, query term coverage), and yield macro-features that improve predictive R^2 from $\sim$0.20 to 0.43–0.47. (3) Treating coalitions as "meta-neurons," we trace their evolution across consecutive layers, finding that most groups vanish or split while only a small fraction persist—supporting the view that deeper MLPs act primarily as feature filters rather than creators. Applied to LLaMA, Mistral, and Pythia LoRA rerankers, we show that hedonic coalitions consistently uncover reproducible and functionally indispensable computational units. To our knowledge, this is the first work to use game theory to identify, validate, and track synergistic neuron groups in fine-tuned LLMs. All code, models, and datasets are provided with the submission.

2 Background

We begin by outlining the fundamentals of hedonic games and their application in modeling cooperative behavior. We then describe the transformer architecture with an emphasis on MLP sublayers.

2.1 Hedonic Games and PAC-Stable Coalition Formation

A *hedonic coalition formation game* (Dreze & Greenberg, 1980) consists of a set of players N who exhibit preferences over groups they might join. Formally, each player $i \in N$ ranks all coalitions $S \subseteq N$ that contain i; in our setting, we assume that players have *cardinal* utilities over coalitions. Given a player $i \in N$ and a coalition S containing i, player i's utility from joining S is $u_i(S) \in \mathbb{R}$, which we later instantiate as a function of i's strongest partners within S.

Our goal is to identify a *coalition structure* or *partition* of the player set which satisfies certain desiderata (Aziz & Savani, 2016). Given a coalition structure π, we let $\pi(i)$ designate the coalition containing player i under π. *Core stability* (Bogomolnaia et al., 2002) is a key cooperative solution concept. We say that a coalition $S \subseteq N$ *blocks* a coalition structure π if every player $i \in S$ strictly prefers S to their assigned coalition $\pi(i)$, i.e., $u_i(S) > u_i(\pi(i))$ for all $i \in S$. A coalition structure π is *core stable* (or simply *stable*) if no blocking coalitions exist.

Enumerating agents' preferences over all coalitions is infeasible; with n agents, each agent needs to express their preferences over 2^{n-1} potential groups. Sliwinski & Zick (2017) propose using *Probably Approximately Correct (PAC)* guarantees (Kearns & Vazirani, 1995; Shashua, 2009). The key insight of this framework is to sample players' preferences rather than utilize complete preferences over all coalitions. Given a distribution D over coalitions, a coalition structure $\hat{\pi}$ is called *ε-PAC stable* if

$$\Pr_{S \sim D}\left[S \text{ core blocks } \hat{\pi}\right] \;\leq\; \varepsilon.$$

Here, D is the distribution over sampled coalitions used to approximate neuron preferences. Intuitively, while it is *possible* that $\hat{\pi}$ is not core stable, the probability of observing a blocking coalition for $\hat{\pi}$ under the distribution D is small.

A PAC stabilization algorithm takes $m = \text{poly}(n, \frac{1}{\varepsilon}, \log \frac{1}{\delta})$ samples from D and outputs an ε-PAC stable partition with probability at least $1 - \delta$. Intuitively, δ captures the probability that the m samples we took are not representative of the 'true' data distribution D.

Top-Responsive Hedonic Games in Neural Networks. We estimate pairwise affinities ϕ_{ij} between "players" (neurons) from weights and co-activations. These affinities allow us to construct a hedonic game in which each neuron evaluates coalitions based on the presence of preferred partners. To capture this behavior, we model the setting as a *top-responsive game*. In a top-responsive game, every player i associates each coalition $S \ni i$ with a unique *choice set* $ch(i, S) \subseteq S$ that represents the subset of partners most important to i. Preferences are then determined entirely by these choice sets: a player prefers one coalition over another if its choice set is ranked higher, and if two coalitions yield the same choice set, the smaller coalition is favored. This restriction makes coalition evaluation tractable, as each neuron only needs to consider its most valued partners rather than all possible groups.

The top responsive framework is flexible, as choice sets may consist of a single strong partner, multiple valued partners, or even subsets selected according to synergy between members. The key requirement is that choice sets are uniquely defined and utilities are represented in an *informative* way, so that distinct choice sets correspond to distinct utility "buckets". Under these conditions, the Top-Covering algorithm (Alcalde & Revilla, 2004; Dimitrov & Sung, 2007) can be applied to efficiently compute an (ε, δ) PAC-stable partition (Sliwinski & Zick, 2017). This enables us to identify groups of neurons that form stable coalitions under the distribution of observed samples. Further details and extensions are provided in Appendix A.1.

2.2 Transformer MLPs and Latent Feature Formation

Each LLM transformer block contains a gated MLP that expands the hidden state, applies a non-linearity, and then projects it back to the model dimension. Let the hidden vector entering the MLP at layer ℓ be $\vec{h} \in \mathbb{R}^{d_{\text{model}}}$, and let $d_{\text{ff}} > d_{\text{model}}$ denote the intermediate width. In LLaMA-3-style architectures Dubey et al. (2024), the computation proceeds as:

$$\vec{z}_{\text{up}} = W_{\text{up}}\vec{h}, \quad \vec{z}_{\text{gate}} = W_{\text{gate}}\vec{h}; \quad \vec{g} = \text{SiLU}(\vec{z}_{\text{gate}}) \odot \vec{z}_{\text{up}}, \quad \vec{h}' = W_{\text{down}}\vec{g}.$$

where $W_{\text{up}}, W_{\text{gate}} \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$ and $W_{\text{down}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$. The element-wise product $\vec{g}$ binds the *gate* signal – which selects or suppresses coarse abstractions – with the *up* signal that carries candidate feature directions. W_{down} then recombines these activated features. During this process, abstract features may *emerge* (via new activation directions), *merge* (when multiple features co-activate), *split* (when previously unified features diverge), or *disappear* (if suppressed by gating) (Elhage et al., 2021; Tian et al., 2023).

LoRA-adapted projections. In our setup, only the MLP projection matrices are fine-tuned using Low-Rank Adaptation (LoRA) (Hu et al., 2022). For any weight matrix $W \in \mathbb{R}^{m \times n}$, LoRA introduces a low-rank update of the form:

$$\tilde{W} = W + \Delta W, \quad \Delta W = \frac{\alpha}{r} AB^{\top}, \tag{1}$$

where $A \in \mathbb{R}^{m \times r}$ and $B \in \mathbb{R}^{n \times r}$ are the learned parameters, r is the rank, and α is a scaling factor. When applied to W_{up} and W_{gate}, we obtain:

$$\vec{z}_{\text{up}} = (W_{\text{up}} + \Delta W_{\text{up}})\vec{h}, \quad \vec{z}_{\text{gate}} = (W_{\text{gate}} + \Delta W_{\text{gate}})\vec{h}.$$

The updates ΔW_{up} and ΔW_{gate} are low-rank, as a result, they introduce only a small set of new feature directions in the high-dimensional MLP space. But because these directions are diffused across neurons, visual inspection of weight updates reveals no obvious structure—leading to our central question: *which subsets of neurons cooperate to encode task-specific behavior under LoRA?*

In Section 3, we develop a game-theoretic framework that directly identifies these functional coalitions, revealing how LoRA's parameter-efficient updates create localized but coordinated changes that encode task-relevant abstractions without requiring exhaustive analysis of all possible neuron combinations.

3 METHODOLOGY

We present a game-theoretic framework to identify and track *latent coalitions*—cooperating groups of neurons within MLP layers of LoRA-tuned transformer models. Our approach consists of two stages: first, we formalize the intra-layer coalition discovery as a game with hedonic utilities and apply the PAC-Top-Cover algorithm to find stable neuron groupings; second, we connect these coalitions across layers using maximum-weight bipartite matching to trace how these abstract computational units evolve through consecutive layers in the network hierarchy.

3.1 PROBLEM STATEMENT: COALITION DISCOVERY AND TRACKING IN TRANSFORMER MLPS

Let L be a transformer-based LLM fine-tuned for a scalar prediction task (e.g., relevance scoring) via LoRA. Let $\ell \in \{1, 2, \ldots, d\}$ denote an MLP layer in the network, where d is the total number of layers, with $n = d_{\text{ff}}$ neurons in its intermediate dimension. Denote the down-projection weight matrix as $W_{\text{down}}^{(\ell)} \in \mathbb{R}^{d_{\text{model}} \times n}$, where each column $[W_{\text{down}}^{(\ell)}]_{\cdot,i}$ represents the learned projection vector for neuron i. Here neuron i, refers to the i^{th} MLP channel in d_{ff}.

Our goal is to identify a partition $\pi^{(\ell)} = \{C_1, C_2, \ldots, C_k\}$ of neurons in layer ℓ such that each subset $C_i \subseteq \{1, \ldots, n\}$ captures a set of neurons that exhibit strong *synergy*—cooperative behavior in forming a semantic unit. We define synergy through a pairwise valuation function ϕ_{ij}. Then, across layers, we aim to match coalitions from $\pi^{(\ell)}$ to those in $\pi^{(\ell+1)}$, enabling us to model feature *persistence*, *splitting*, *merging*, and other dynamic events.

3.2 CONSTRUCTING PAIRWISE VALUATIONS AND UTILITY SCORES

PAC-Top-Cover uses samples of coalitions $S \sim D$ to estimate each agent's top-k choice set within the remaining pool. We first compute pairwise valuations ϕ_{ij}, which quantify affinity or synergy between neurons. We instantiate two complementary valuation functions:

Orthogonal-Co-Activation (OCA). This approach combines two intuitions: neurons with orthogonal weight vectors may capture complementary features, while neurons with high activation correlation may process similar patterns. For neuron pair (i, j), we define:

$$\phi_{\text{OCA}}(i,j) = \big(1 - |\cos(W_i, W_j)|\big)\ \rho(a_i, a_j), \quad \rho(a_i, a_j) = \frac{\text{Cov}[a_i, a_j]}{\sigma_i \sigma_j}$$

where W_i is the i-th column of $W_{\text{down}}^{(\ell)}$ (neuron i's output weights), and a_i denotes neuron i's activations. The cosine term favors pairs with dissimilar weight vectors, while the correlation term captures their collaborative activation patterns (Pearson's correlation).

Pairwise Ablation Synergy (PAS). To directly measure the synergistic interaction between neurons i and j, we compute the second-order interaction effect through ablation. Let $\ell(x)$ denote the model's logit output[1] for input x, and $\ell_{-S}(x)$ denote the logit when neurons in set S are ablated (set to their pre-LoRA weight). The true interaction between neurons i and j is:

$$\phi_{\text{PAS}}(i,j) = -\mathbb{E}_{x \sim \mathcal{D}} \left[\ell_{-\{i,j\}}(x) - \ell_{-i}(x) - \ell_{-j}(x) + \ell(x) \right].$$

This measures how the joint ablation of both neurons differs from the sum of individual ablations. For computational efficiency with large n, we approximate this using gradient computations:

$$\phi_{\text{PAS}}(i,j) \approx -\frac{\partial^2 \ell}{\partial a_i \partial a_j} \cdot \mathbb{E}[a_i a_j],$$

where the mixed partial derivative captures the interaction between neuron activations.

[1] We use a *layer-local logit* $\ell^{(\ell)}(x) = w^\top h'^{(\ell)}(x) + b$, i.e. the scalar score obtained *immediately after* the layer-ℓ MLP (including residual addition) but *before* entering block $\ell + 1$. Our goal is to discover coalitions that are intrinsically synergistic at the point they are formed. w, b are cloned from the final task head and kept fixed for all layers. For readability we drop the superscript when the layer is clear from context.

We experiment with both OCA (structural heuristic) and PAS (functional ablation-based) valuations to test robustness of our framework. In both the above defined pairwise valuation functions, positive values indicate *synergy* (neurons cooperate to produce information neither could alone), while negative values indicate *redundancy* (neurons provide overlapping information). We now use these valuation functions, to compute choice sets, which is used to compute the utility of a neuron in a set that is used by the PAC Top-Cover algorithm.

Multi-Friend Choice Sets (MFC). In this formation, each neuron is allowed to anchor its preference not on a single partner but on a *set of top-k partners*. For player i, the choice set within coalition S is

$$Ch(i, S) = \arg \max_{\substack{T \subseteq S \setminus \{i\} \\ |T|=k}} \sum_{j \in T} \phi_{ij},$$

with ties broken deterministically to ensure uniqueness. Utilities are then defined as $u_i(S) = \frac{1}{k} \sum_{j \in Ch(i,S)} \phi_{ij}$. This normalized model captures *multi-partner synergy*, where a neuron's activation is meaningful only when several complementary features are present. We refer to this algorithmic instantiation as **Hedonic-MFC**.

3.3 The PAC Top-Cover algorithm.

The PAC Top-Cover algorithm (Sliwinski & Zick, 2017; Alcalde & Revilla, 2004) provides an efficient way to identify stable coalitions of neurons under top-k preferences. The top-k variant of this algorithm allows every neuron i to nominate up to k partners within sampled coalitions, based on the highest affinity scores ϕ_{ij}. In each round, the algorithm samples a batch of candidate coalitions (with sizes constrained to lie between $k_{\min}$ and $k_{\max}$), constructs choice sets B_i for all neurons in the active pool R, and builds a directed preference graph where edges $i \to j$ represent top-k selections. Here B_i denotes the estimated top-k choice set for neuron i, i.e., the subset of partners that maximize its utility under the current sampled coalitions, computed via the MFC rule introduced in Section 3.2. Stable coalitions are then extracted as sink strongly connected components that are also closed under these choice sets. Removing each coalition from R and repeating yields a full partition of the neurons. The algorithm is detailed in Appendix C.

The PAC guarantee ensures that with $O(n^2 \varepsilon^{-1} \log(n/\delta))$ samples per round, the resulting partition is ε-approximately stable with probability at least $1 - \delta$. This provides theoretical backing that the discovered coalitions capture robust cooperative structure among neurons. We next ask how the coalitions identified at one layer relate to those in subsequent layers.

3.4 Tracking Coalitions Across Layers

Our hypothesis is that coalitions capture intermediate features that may *persist*, *merge*, *split*, or *disappear* as computation proceeds through the network. Tracking such transitions provides an exploratory view of how features evolve across depth.

For each pair of coalitions (C, C') from consecutive layers ℓ and $\ell + 1$, we measure their *interaction mass*, which serves as a heuristic to quantify how strongly one coalition influences the next:

$$M(C, C') \;=\; \frac{1}{|C| \cdot |C'|} \sum_{p \in C} \sum_{q \in C'} \Big(|W_{\text{up}}^{(\ell+1)}[q, p]| + |W_{\text{gate}}^{(\ell+1)}[q, p]| \Big) \cdot A_p,$$

where $W_{\text{up}}^{(\ell+1)}, W_{\text{gate}}^{(\ell+1)} \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$ are the LoRA-adapted projection matrices of layer $\ell + 1$, $p \in \{1, \ldots, d_{\text{ff}}\}$ indexes source neurons from layer ℓ, $q \in \{1, \ldots, d_{\text{ff}}\}$ indexes target neurons in layer $\ell+1$, and $A_p = \mathbb{E}_x[|a_p^{(\ell)}(x)|]$ is the mean absolute activation of neuron p over the training distribution. This formulation captures both the additive (W_{up}) and multiplicative gating ($W_{\text{gate}} \times$ SiLU) pathways, while normalizing by coalition sizes ensures comparability across widths. We assemble the interaction masses into a bipartite matrix and solve a maximum-weight matching problem to align coalitions across layers. For each match, we compute the fraction of a source coalition's output that flows into a target (α) and the fraction of a target's input originating from that source (β). These ratios allow us to classify transitions into persistence (both high), splitting (low α, high β), merging (high α, low β), or disappearance (both low).

We stress that this analysis is exploratory. Transformers have residual connections, so neurons at layer ℓ influence all deeper layers, not just $\ell + 1$. Our method only captures local dynamics and likely underestimates long-range interactions, but it offers a first step toward visualizing how abstract feature groups may evolve through the network.

4 EXPERIMENTS

We empirically validate our framework on three LLM architectures and three scalar-output IR tasks. We first describe models, tasks, and baselines, then present evaluation protocols and results (Tables 1, 2, 3, Appendix 6).

Models. We study LLaMA-3.1-8B (Dubey et al., 2024), Mistral-7B-v0.1 (Jiang et al., 2023), and Pythia-6.9B (Biderman et al., 2023), each adapted via LoRA (rank $r = 8$) restricted to MLP layers 7–14. Preliminary analysis showed these layers carry the strongest task-specific activity (Nijasure et al., 2025). Fine-tuning uses AdamW ($\eta = 2 \times 10^{-4}$, batch size 128, 3 epochs), with all base weights frozen. Performance of these fine-tuned LoRA models is further documented in Appendix F.

Tasks. Tasks are scalar objectives defined over query–document pairs from MS MARCO (Bajaj et al., 2018): (1) Covered-Query-Term Ratio (*CQTR*) = fraction of query terms present in the document, (2) Mean of Stream-Length Normalized Term Frequency (*Mean-TF/L*) = mean of length-normalized term frequencies, (3) Relevance Modelling (*RM*) = supervised passage ranking. CQTR and Mean-TF/L use MSE loss, RM uses NDCG. Models are trained on 500k pairs, validated on 5k, and evaluated OOD on TREC DL-19/20.

Baselines. We compare five coalition builders: *Random* (uniform neuron subsets with matched size histogram), *K-means (Spherical)* (on ℓ_2-normalized mean activations, k matched to Hedonic partition), *Hierarchical (Ward+cos)* (agglomerative under cosine distance, cut at same k), *Hedonic-OCA* (PAC-Top-Cover with ϕ_{OCA}), *Hedonic-PAS* (PAC-Top-Cover with ϕ_{PAS}).

For Hedonic sampling we draw $m = 8{\times}10^5$ candidate coalitions (size $[2, 10]$), retain top $\omega = 8{\times}10^4$ by utility, and use $\varepsilon = \delta = 0.1$. Choice sets use top-3 partners. Cross-layer matching uses thresholds $(\alpha_{\text{hi}}, \alpha_{\text{lo}}) = (0.7, 0.1)$ tuned on a 1% held-out split. All methods run on 4×A100-80GB GPUs; PAC-Top-Cover completes in 90 min (OCA) and 280 min (PAS). All numbers are averaged over 3 seeds with 95% confidence intervals.

Evaluation. We first report intrinsic synergy metrics (Appendix G) as diagnostics, then evaluate coalitions extrinsically with three tests:

- *OOD Drop.* For coalition C, we measure the performance drop on $\mathcal{D}_{\text{OOD}}$ (DL-19/20) when C is ablated (neurons reset to pre-LoRA weights):
$$\Delta\mathcal{M}(C) = \mathcal{M}(\{\ell(x)\}) - \mathcal{M}(\{\ell_{-C}(x)\}),$$
where $\mathcal{M}$ is NDCG@10 for RM and $-$MSE for CQTR/Mean-TF/L. Larger $\Delta\mathcal{M}(C)$ indicates greater functional importance.
- *Feature Alignment.* Each coalition's mean activation $a_C(x)$ is compared with known IR heuristics (list of MSLR features Qin & Liu (2013b)). Alignment is defined as the maximum squared Pearson correlation:
$$R^2(C) = \max_j \ \text{Corr}^2(a_C(x), f_j(x)).$$
- *Coalition Predictivity.* Coalitions are treated as macro-features $A(x) \in \mathbb{R}^k$. A ridge regression $\hat{y}(x) = w^\top A(x)$ is trained on MS MARCO and evaluated OOD; we report R^2 for RM, CQTR, and Mean-TF/L.

Next, we discuss the results reported in Tables 1 (extrinsic coalition evaluation), Table 2(coalition predictivity), Table 3 (coalition transfer dynamics) and Appendix Table 6 (intrinsic coalition evaluation).

Experimental Results.

Table 1: Extrinsic Evaluation: OOD Drop (↑) and Feature Alignment R^2 (↑) on DL-19/20. Mean $\pm 95\%$ CI across three seeds. Larger values indicate more functionally important and interpretable coalitions.

Task / Algorithm	LLaMA-3.1		Mistral		Pythia	
	OOD Drop	Align R^2	OOD Drop	Align R^2	OOD Drop	Align R^2
Covered Query Term Ratio						
Random	0.01 ± 0.01	0.05 ± 0.02	0.00 ± 0.01	0.06 ± 0.02	0.01 ± 0.01	0.05 ± 0.02
K-means	0.02 ± 0.01	0.12 ± 0.02	0.03 ± 0.01	0.13 ± 0.02	0.02 ± 0.01	0.11 ± 0.02
Hier. clustering	0.03 ± 0.01	0.15 ± 0.02	0.03 ± 0.01	0.16 ± 0.02	0.03 ± 0.01	0.14 ± 0.02
Hedonic (OCA)	0.07 ± 0.01	0.41 ± 0.02	0.09 ± 0.01	0.44 ± 0.02	0.08 ± 0.01	0.45 ± 0.02
Hedonic (PAS)	$\mathbf{0.11} \pm 0.005$	$\mathbf{0.58} \pm 0.01$	$\mathbf{0.13} \pm 0.006$	$\mathbf{0.61} \pm 0.01$	$\mathbf{0.14} \pm 0.006$	$\mathbf{0.63} \pm 0.01$
Mean of Normalized Term Frequency						
Random	0.01 ± 0.01	0.04 ± 0.02	0.01 ± 0.01	0.05 ± 0.02	0.01 ± 0.01	0.05 ± 0.02
K-means	0.02 ± 0.01	0.11 ± 0.02	0.02 ± 0.01	0.12 ± 0.02	0.02 ± 0.01	0.10 ± 0.02
Hier. clustering	0.03 ± 0.01	0.14 ± 0.02	0.03 ± 0.01	0.15 ± 0.02	0.03 ± 0.01	0.13 ± 0.02
Hedonic (OCA)	0.06 ± 0.01	0.38 ± 0.02	0.08 ± 0.01	0.41 ± 0.02	0.07 ± 0.01	0.40 ± 0.02
Hedonic (PAS)	$\mathbf{0.10} \pm 0.005$	$\mathbf{0.55} \pm 0.01$	$\mathbf{0.12} \pm 0.006$	$\mathbf{0.59} \pm 0.01$	$\mathbf{0.13} \pm 0.006$	$\mathbf{0.60} \pm 0.01$
Relevance Modelling						
Random	0.02 ± 0.01	0.06 ± 0.02	0.01 ± 0.01	0.07 ± 0.02	0.02 ± 0.01	0.06 ± 0.02
K-means	0.03 ± 0.01	0.13 ± 0.02	0.04 ± 0.01	0.14 ± 0.02	0.03 ± 0.01	0.13 ± 0.02
Hier. clustering	0.04 ± 0.01	0.17 ± 0.02	0.04 ± 0.01	0.18 ± 0.02	0.04 ± 0.01	0.16 ± 0.02
Hedonic (OCA)	0.09 ± 0.01	0.47 ± 0.02	0.10 ± 0.01	0.49 ± 0.02	0.09 ± 0.01	0.48 ± 0.02
Hedonic (PAS)	$\mathbf{0.14} \pm 0.006$	$\mathbf{0.63} \pm 0.01$	$\mathbf{0.16} \pm 0.006$	$\mathbf{0.65} \pm 0.01$	$\mathbf{0.17} \pm 0.007$	$\mathbf{0.67} \pm 0.01$

Table 2: Coalition Predictivity (R^2 on OOD sets DL-19/20), averaged across three LLMs (LLaMA-3.1, Mistral, Pythia). Coalitions are used as macro-features in ridge regression trained on MS MARCO. Hedonic coalitions yield substantially higher R^2 than clustering or random baselines.

Algorithm	CQTR	Mean-TF/L	Relevance (RM)
Random	0.08 ± 0.02	0.09 ± 0.02	0.12 ± 0.02
K-means	0.16 ± 0.01	0.15 ± 0.01	0.21 ± 0.01
Hier. clustering	0.18 ± 0.01	0.17 ± 0.01	0.21 ± 0.01
Hedonic (OCA)	0.34 ± 0.01	0.33 ± 0.01	0.38 ± 0.01
Hedonic (PAS)	$\mathbf{0.43} \pm 0.008$	$\mathbf{0.42} \pm 0.008$	$\mathbf{0.47} \pm 0.008$

Functional importance and interpretability (Table 1). Across all three models and tasks, hedonic coalitions are markedly more *causal* and *interpretable* than clustering or random partitions. Ablating a single hedonic coalition (ablation = restoring those neurons to their pre-LoRA state) yields the largest OOD performance drops: for CQTR on LLaMA/Mistral/Pythia the OOD drop rises from ≈0.02–0.03 (K-means/Hier.) to 0.11–0.14 with Hedonic-PAS—about a **3–5**× increase; similar gaps hold for Mean-TF/L (0.10–0.13 vs. 0.02–0.03) and RM (0.14–0.17 vs. 0.03–0.04). At the same time, coalition activations align far more strongly with IR heuristics: alignment R^2 climbs from ∼0.11–0.18 (clustering) to **0.55–0.67** (Hedonic-PAS), with Hedonic-OCA consistently second-best (≈0.38–0.49). Confidence intervals are narrow throughout, indicating stable estimates over seeds. Taken together, these results show that hedonic coalitions are both **functionally indispensable**—their removal produces large OOD degradation—and **semantically grounded**, tracking BM25/IDF/coverage signals far better than baselines.

Predictive macro-features (Table 2). Treating each coalition as a macro-feature and training a ridge regressor on MS MARCO, we see large generalization gains on DL-19/20. Averaged over LLaMA, Mistral, and Pythia, Hedonic-PAS attains $R^2 =$ **0.43/0.42/0.47** on CQTR/Mean-TF/L/RM, roughly **2–3**× higher than K-means/Hier. (≈0.15–0.21) and far above Random (≈0.08–0.12). Hedonic-OCA also performs strongly (≈0.33–0.38), reinforcing the pattern from the extrinsic ablations: utilities that respect *synergy* (PAS) or *partner preference* (OCA) produce coalitions that behave like **robust, transferable features**, not just co-activation clusters. This bridges intrinsic synergy to downstream utility: coalitions that score high on synergy also yield higher OOD predictivity.

Table 3: Dynamics of coalitions across layers 7–14 for three tasks. Each cell shows percentage of coalitions exhibiting the event relative to all coalitions present in the *source* layer (except *merge*).

	Mistral				LLaMA				Pythia			
Layer →	Persist	Merge	Split	Vanish	Persist	Merge	Split	Vanish	Persist	Merge	Split	Vanish
Covered Query Term Ratio												
7 → 8	12.1%	0.0%	28.9%	59.0%	3.2%	0.0%	35.4%	61.4%	7.8%	0.0%	31.9%	60.3%
8 → 9	4.8%	0.0%	38.4%	56.8%	5.1%	0.0%	28.6%	66.3%	4.9%	0.0%	33.7%	61.4%
9 → 10	6.2%	0.0%	31.5%	62.3%	4.8%	0.0%	30.2%	65.0%	5.5%	0.0%	30.8%	63.7%
10 → 11	3.8%	0.0%	29.7%	66.5%	7.9%	0.0%	32.1%	60.0%	5.9%	0.0%	30.9%	63.2%
11 → 12	4.2%	0.0%	27.8%	68.0%	3.5%	0.0%	29.8%	66.7%	3.9%	0.0%	28.8%	67.3%
12 → 13	11.3%	0.0%	30.2%	58.5%	6.8%	0.0%	27.4%	65.8%	9.1%	0.0%	28.8%	62.1%
13 → 14	10.5%	0.0%	24.3%	65.2%	7.2%	0.0%	23.1%	69.7%	8.9%	0.0%	23.7%	67.4%
Stream Length Normalized Term Frequency												
7 → 8	6.4%	0.5%	35.8%	57.3%	2.1%	0.0%	19.7%	78.2%	4.3%	0.3%	28.4%	67.0%
8 → 9	1.8%	0.2%	51.2%	46.8%	3.8%	0.1%	20.3%	75.8%	2.8%	0.1%	36.2%	60.9%
9 → 10	2.9%	0.1%	23.1%	73.9%	3.4%	0.0%	22.8%	73.8%	3.2%	0.0%	22.9%	73.9%
10 → 11	1.3%	0.3%	23.7%	74.7%	5.9%	0.2%	25.5%	68.4%	3.6%	0.2%	24.6%	71.6%
11 → 12	1.2%	0.1%	22.9%	75.8%	1.7%	0.0%	24.1%	74.2%	1.4%	0.0%	23.5%	75.1%
12 → 13	6.3%	0.4%	37.8%	55.5%	3.2%	0.1%	19.8%	76.9%	4.8%	0.2%	29.3%	65.7%
13 → 14	7.1%	0.2%	19.5%	73.2%	4.7%	0.0%	17.1%	78.2%	5.9%	0.1%	18.3%	75.7%
Relevance												
7 → 8	8.2%	0.0%	32.7%	59.2%	1.5%	0.0%	22.4%	76.1%	5.2%	0.0%	28.1%	66.7%
8 → 9	2.1%	0.0%	46.8%	51.1%	3.5%	0.0%	22.8%	73.7%	2.8%	0.0%	35.3%	61.9%
9 → 10	3.5%	0.0%	26.3%	70.2%	3.9%	0.0%	25.5%	70.6%	3.7%	0.0%	25.9%	70.4%
10 → 11	2.0%	0.0%	26.5%	71.4%	6.7%	0.0%	28.3%	65.0%	4.2%	0.0%	27.4%	68.4%
11 → 12	1.7%	0.0%	25.4%	72.9%	2.0%	0.0%	26.5%	71.4%	1.9%	0.0%	25.9%	72.2%
12 → 13	8.0%	0.0%	34.0%	58.0%	4.0%	0.0%	22.0%	74.0%	6.1%	0.0%	28.2%	65.7%
13 → 14	8.7%	0.0%	21.7%	69.6%	5.4%	0.0%	18.9%	75.7%	7.1%	0.0%	20.3%	72.6%

Coalition dynamics across depth (Table 3). Across layers 7→14, three trends are consistent: (i) *vanish dominates* (typically *60–75%* of coalitions disappear at the next layer), indicating downstream MLPs act as *filters/refiners* rather than combiners; (ii) *splits are common* (≈20–50%, depending on task/layer), suggesting feature *refinement* is more prevalent than wholesale reuse; and (iii) *merges are near-zero*, implying whole motifs are rarely recomposed from separate groups. Persistence is generally low ($<\sim$12%), with a mild *delayed persistence uptick* around 12→13 for CQTR and RM (≈8–11%), echoing a "late stabilization" phase. Mean-TF/L exhibits the strongest pruning (vanish $>$70% across several transitions), consistent with simple frequency statistics being isolated early and aggressively culled later. These dynamics support our central claim: **cooperative units are formed, then predominantly pruned or refined rather than fused**, aligning with the heavy-tailed coalition sizes and the functional importance patterns observed above.

5 Discussion

SAEs vs Hedonic Neurons. Sparse Autoencoders (SAEs) (Huben et al., 2024) uncover interpretable features by learning sparse dictionaries that reconstruct activations and disentangle polysemantic units. In contrast, our framework keeps neurons as primitives and asks how they cooperate. By modeling them as agents in a hedonic game, we capture nonlinear synergies: coalitions whose joint ablation impacts behavior beyond the sum of parts. Unlike SAEs, which re-express activation space, hedonic coalitions are grounded in weight geometry and preference structure, surfacing cooperative "wiring-level" units already encoded in the parameters. The two approaches are complementary: SAEs expose monosemantic features, while hedonic analysis highlights how neurons collaborate to realize them.

Coalition size distribution. Coalition sizes follow a heavy-tailed Zipfian law: each layer contains a few large "macro" groups, mid-sized units, and many size-2 specialists, resembling vocabulary statistics in language. Disappearance rates rise after layer 12, suggesting deeper MLP blocks act more as feature filters than creators. Together, these findings imply that hedonic coalitions are natural computational units shaped by training dynamics—early layers construct rich representations, while later ones selectively retain task-relevant features.

6 Related Work

Mechanistic interpretability of transformer LLMs has focused on understanding both individual neurons and structured groups. Geva et al. (2021) showed that feed-forward layers act as key–value memories, with neurons detecting input patterns (keys) and injecting values into the representation. Dai et al. (2022) identified "knowledge neurons" in MLPs that encode factual associations, demonstrating that small groups of neurons can robustly store discrete knowledge.

Beyond single-neuron analysis, Bricken et al. (2023) applied dictionary learning to extract sparse, interpretable features from polysemantic activations. Balagansky et al. (2025) tracked feature persistence and merging across layers, complementing our coalition-evolution view. Sparse probing (Gurnee et al., 2023) further revealed that early layers are highly polysemantic while deeper layers specialize, underscoring the need to model neuron groups and their dynamics. Weight-based methods also contribute: Davies (2025) decoded neuron weights into semantic concepts, while Pearce et al. (2024) and Bushnaq et al. (2025) developed direct weight-space feature discovery.

While hedonic games have rarely been explored in interpretability, Koulali and Koulali (Koulali & Koulali, 2023) showed their utility for feature selection, providing theoretical foundations for our approach. Our work extends these lines by explicitly framing neuron collaboration as a hedonic game, enabling principled discovery and tracking of *stable coalitions* that serve as latent computational units in transformer MLPs.

7 Conclusion, Limitations and Future Work

We introduced **Hedonic Neurons**, a game-theoretic framework that models neurons in transformer MLPs as players in a top-responsive hedonic game. Using the PAC-Top-Cover algorithm with correlation-based (OCA) or ablation-based (PAS) valuations, we identified stable coalitions that capture cooperative structure beyond what clustering can reveal. Across three LLM architectures and scalar IR tasks, hedonic coalitions achieve average improvements of +0.29 *Pairwise* and +0.49 *Ratio* synergy over the strongest baseline, while extrinsic evaluations show they are functionally indispensable: ablations yield 3–5$\times$ larger OOD performance drops, alignment with IR heuristics rises from $\sim$0.15 to 0.55–0.67, and predictive R^2 improves from $\sim$0.20 to 0.43–0.47. Coalition dynamics further reveal that most groups vanish or split across depth, with merges rare and persistence limited, supporting the view that MLPs act primarily as filters and refiners of features.

Our approach has limitations: utilities depend on layer-local logits and second-order ablations, omitting higher-order interactions and attention mechanisms, and the current formulation yields disjoint coalitions despite early-layer polysemy. Future work will extend to overlapping coalitions via fractional hedonic games, integrate attention heads for joint sub-module analysis, and design low-variance estimators to reduce $O(n^2)$ ablation costs. Coupling hedonic discovery with concept-activation vectors may also yield interpretable primitives aligned with human-understandable features. Taken together, HedonicNeurons provides a principled foundation for uncovering how cooperative computational units emerge, evolve, and specialize in large-scale language models.

8 Reproducibility Statement

We provide all resources necessary to reproduce our experiments. We make our fine-tuned reranker checkpoints for Pythia, Mistral, and LLaMA3 models available on HuggingFace (see supplementary material). The training dataset (Tevatron MSMARCO Passage Augmented) and evaluation dataset (TREC DL 2019) are publicly available, with preprocessing steps following the Tevatron MSMARCO implementation. All scripts used for coalition generation, partitioning, clustering baselines, and evaluation are included in the repository, along with deepspeed configuration files for finetuning. Coalition files (`.pkl`) and visualization outputs (Sankey plots) are also provided. Together, these resources ensure that the models, tasks, and coalition analyses can be wholly reproduced.

References

José Alcalde and Pablo Revilla. Researching with whom? stability and manipulation. *Journal of Mathematical Economics*, Vol 40(Issue 8):pp. 869–887, 2004.

Haris Aziz and Rahul Savani. Hedonic games. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (eds.), *Handbook of Computational Social Choice*, chapter 15. Cambridge University Press, 2016.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension dataset, 2018.

Nikita Balagansky, Ian Maksimov, and Daniil Gavrilov. Mechanistic permutability: Match features across layers. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, pp. 2397–2430, 2023.

Anna Bogomolnaia, , and Matthew O. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, Vol 38(Issue 2):pp. 201–230, 2002.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. 2023.

Lucius Bushnaq, Dan Braun, and Lee Sharkey. Stochastic parameter decomposition. *arXiv preprint arXiv:2506.20790*, 2025.

Tue Minh Cao, Nhat Hoang-Xuan, Hieu Pham, Phi Le Nguyen, and My T. Thai. Neurflow: Interpreting neural networks through neuron groups and functional interactions. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Tanya Chowdhury, Atharva Nijasure, and James Allan. Probing ranking llms: A mechanistic analysis for information retrieval. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, pp. 336–346, 2025.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. Overview of the trec 2019 deep learning track. *In arXiv, doi: ArXiv:2003.07820*, 2020.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 8493–8502, 2022.

Harry J Davies. Decoding specialised feature neurons in llms with the final projection layer. *In arXiv, doi ArXiv:2501.02688*, 2025.

Dinko Dimitrov and Shao Chin Sung. On top responsiveness and strict core stability. *Journal of Mathematical Economics*, Vol 43(Issue 2):pp. 130–134, 2007.

Jacques H Dreze and Joseph Greenberg. Hedonic coalitions: Optimality and stability. *Econometrica: Journal of the Econometric Society*, pp. 987–1003, 1980.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *In arXiv, doi: ArXiv:2407.21783*, 2024.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, Vol 1(Issue 1):pp. 12, 2021.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *In arXiv, doi ArXiv:2209.10652*, 2022.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. Scaling deep contrastive learning batch size under memory limited setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP*, 2021.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5484–5495, 2021.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research (TMLR)*, 2023. ISSN 2835-8856.

John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pp. 4129–4138, 2019.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Michael J Kearns and Umesh V Vazirani. Computational learning theory. *Association for Computing Machinery (ACM) SIGACT News*, Vol 26(Issue 1):pp. 43–45, 1995.

Rim Koulali and Mohammed-Amine Koulali. Feature selection as a hedonic coalition formation game for arabic topic detection. *Pattern Recognition Letters*, Vol 172:137–143, 2023.

Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 120–127, 2001.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 2421–2425, 2024a.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 2421–2425. Association for Computing Machinery, 2024b.

Xueguang Ma, Luyu Gao, Shengyao Zhuang, Jiaqi Samantha Zhan, Jamie Callan, and Jimmy Lin. Tevatron 2.0: Unified document retrieval toolkit across scale, language, and modality. *In arXiv, doi ArXiv:2505.02466*, 2025.

Atharva Nijasure, Tanya Chowdhury, and James Allan. How relevance emerges: Interpreting lora fine-tuning in reranking llms. *In arXiv, doi: ArXiv:2504.08780*, 2025.

Michael T Pearce, Thomas Dooms, Alice Rigg, Jose M Oramas, and Lee Sharkey. Bilinear mlps enable weight-based mechanistic interpretability. *arXiv preprint arXiv:2410.08417*, 2024.

Tao Qin and Tie-Yan Liu. Introducing LETOR 4.0 datasets. *CoRR*, Vol abs/1306.2597, 2013a.

Tao Qin and Tie-Yan Liu. Introducing letor 4.0 datasets. *arXiv preprint arXiv:1306.2597*, 2013b.

Amnon Shashua. Introduction to machine learning: Class notes 67577. *In arXiv, doi: ArXiv:0904.3664*, 2009.

Jakub Sliwinski and Yair Zick. Learning hedonic games. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, (IJCAI-17)*, pp. 2730–2736, 2017.

Ran Song, Shizhu He, Shuting Jiang, Yantuan Xian, Shengxiang Gao, Kang Liu, and Zhengtao Yu. Does large language model contain task-specific neurons? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7101–7113, 2024.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, Vol 28(Issue 1):pp. 11–21, 1972.

Tevatron. Ms marco augmented dataset, 2024.

Yuandong Tian, Yiping Wang, Beidi Chen, and Simon Shaolei Du. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Xiongtao Zhou, Jie He, Yuhua Ke, Guangyao Zhu, Victor Gutierrez Basulto, and Jeff Pan. An empirical study on parameter-efficient fine-tuning for multimodal large language models. In *Findings of the Association for Computational Linguistics: (ACL)*, pp. 10057–10084, 2024.

A Hedonic Games Prelimineries and PAC Top Cover Intuition

A.1 Hedonic Games

A *hedonic game* (Dreze & Greenberg, 1980) is defined by a finite set of players $N = \{1, \dots, n\}$ and, for each player i, a complete and transitive preference relation $\succ_i$ over the set $\mathcal{N}_i = \{S \subseteq N \mid i \in S\}$ of coalitions that contain i. A *coalition structure* (or *partition*) is a set $\pi = \{C_1, \dots, C_k\}$ of disjoint non-empty coalitions whose union equals N. Throughout this appendix we assume that preferences are given by real-valued utilities $v_i : \mathcal{N}_i \to \mathbb{R}$ so that $S \succ_i T \;\Leftrightarrow\; v_i(S) > v_i(T)$.[2]

A.2 Core Stability

Given a partition π and a coalition $S \subseteq N$, we say that S *blocks* π if every $i \in S$ strictly prefers S to her coalition in π, i.e. $S \succ_i \pi(i)$. A partition is *core-stable* (or simply *in the core*) if it is not blocked by any coalition. Core stability captures the idea that no subset of players has a joint incentive to deviate.

A.3 Why Full Preference Learning Is Infeasible

Precisely learning all utilities $v_i(S)$ is unrealistic because the number of coalitions grows exponentially ($|\mathcal{N}_i| = 2^{n-1}$). Even if we could query any coalition, the sample complexity implied by the pseudo-dimension of general hedonic games is super-polynomial (Proposition 4.9 in (Sliwinski & Zick, 2017)). Hence, any practical method must settle for *approximate* stability based on samples rather than complete preference elicitation.

A.4 PAC-Learning Framework for Hedonic Games

Following (Sliwinski & Zick, 2017), let D be an unknown but fixed distribution over coalitions. A partition π is *ε-PAC stable under D* if

$$\Pr_{S \sim D}\big[S \text{ blocks } \pi\big] \;<\; \varepsilon.$$

An algorithm A *PAC-stabilises* a class $\mathcal{H}$ of hedonic games if, for any game $G \in \mathcal{H}$, distribution D, and parameters (ε, δ), A outputs—with probability at least $1 - \delta$—an ε-PAC-stable partition using a number of samples polynomial in $(n, 1/\varepsilon, \log(1/\delta))$.

A.5 Intuition Behind the Top-Cover Algorithm

Under *additively separable* utilities ($v_i(S) = \sum_{j \in S \setminus \{i\}} u_{ij}$), players exhibit *top-responsiveness*: their evaluation of a coalition is determined by the "best" members plus a size penalty (Alcalde & Revilla, 2004). Top-Cover exploits this property iteratively:

(i) using samples, approximate each player's most preferred subset within the current residual set,

(ii) build directed edges from each player to the members of that subset,

(iii) extract a strongly connected component of minimal size, form it as a coalition, and remove it,

(iv) repeat until all players are assigned.

Each extracted coalition is unlikely to be blocked because every member already sees its best attainable partners within it with high probability.

[2] See Section 2 of (Sliwinski & Zick, 2017) for an extensive discussion of numeric versus ordinal representations.

A.6 Additive Separability Implies Top-Responsiveness

In an additively separable game, for any player i and coalitions $S, T \ni i$,

$$v_i(S) \;>\; v_i(T) \iff \big(\exists j \in S \setminus \{i\} : u_{ij} > u_{ik}\ \forall k \in T \setminus \{i\}\big) \text{ or } \big(S \supset T \wedge v_i(S) = v_i(T)\big).$$

Hence each coalition can be ranked by (a) the highest-valued partner of i (*choice set*) and, if equal, (b) coalition size—the definition of top responsiveness (Alcalde & Revilla, 2004). Consequently, additively separable utilities allow Top-Cover (and its PAC variant) to guarantee an ε-PAC-stable partition.

A.7 Application of Hedonic Games to Neural Networks.

Neurons in a transformer predominantly interact with a limited set of peers—those with highly correlated activations or complementary weights. Treating neurons as players whose utilities are derived from such local synergies fits the additive model naturally. Sampling mini-batches of logits/activations supplies the coalitions needed by the PAC framework, letting us recover *approximately core-stable neuron groups* without exhaustively testing all neuron subsets.

B MAKING ADDITIVE UTILITIES TOP-RESPONSIVE

Notation recap. For each ordered pair of distinct neurons (i, j) we have a *pairwise synergy score* $\phi_{ij} \in \mathbb{R}$ (either ϕ_{OCA} or ϕ_{PAS}; see §3.2). Write Φ for the $n \times n$ matrix with zeros on the diagonal.

B.1 FROM ADDITIVE SCORES TO TOP-RESPONSIVE PREFERENCES

Max-partner utility : Fix a global parameter $k \geq 1$. For a coalition $S \subseteq N$ that contains player i let

$$\texttt{Top}_k(i, S) \;=\; \operatorname*{arg\,max}_{\substack{T \subseteq S \setminus \{i\} \\ |T| \leq k}} \sum_{j \in T} \phi_{ij}.$$

We define

$$u_i(S) \;=\; \sum_{j \in \texttt{Top}_k(i,S)} \phi_{ij}, \quad \text{and} \quad \mathcal{C}_i(S) \;=\; \texttt{Top}_k(i, S).$$

When $k = 1$ this reduces to the familiar "best-friend" utility $u_i(S) = \max_{j \in S \setminus \{i\}} \phi_{ij}$.

Lemma B.1 (Top-responsiveness)**.** *For every player i the preference relation $\succeq_i$ induced by u_i is* top-responsive*: for any two coalitions S, T that contain i*

$$\mathcal{C}_i(S) \succ_i \mathcal{C}_i(T) \implies S \succ_i T.$$

Proof. Let S, T contain i and assume $\mathcal{C}_i(S) \succ_i \mathcal{C}_i(T)$, i.e. $u_i(\mathcal{C}_i(S)) > u_i(\mathcal{C}_i(T))$. Because u_i is *monotone* in the sense that enlarging a set never decreases its utility,[3] we have $u_i(S) \geq u_i(\mathcal{C}_i(S))$ and $u_i(T) = u_i(\mathcal{C}_i(T))$. Hence $u_i(S) > u_i(T)$, so $S \succ_i T$. □

Lemma B.2 (Informative representation)**.** *Given the matrix Φ one can compute $\mathcal{C}_i(S)$ (and therefore the induced ranking) in $O(k\,|S|)$ time. Hence the utility representation is* informative *in the sense of Sliwinski and Zick(Sliwinski & Zick, 2017).*

Proof. $\texttt{Top}_k(i, S)$ requires sorting at most $|S| - 1$ real numbers $\{\phi_{ij}\}_{j \in S \setminus \{i\}}$; the k largest can be found in the stated time using a partial-selection routine. □

Theorem B.3 (Applicability of PAC-Top-Cover)**.** *With utilities u_i from Definition B.1 the induced hedonic game is top-responsive and informative. Consequently, Algorithm* **??** *outputs an ε-PAC-stable partition with probability $1 - \delta$ using $m = \text{poly}\big(n, \frac{1}{\varepsilon}, \log \frac{1}{\delta}\big)$ samples, exactly as in (Sliwinski & Zick, 2017).*

Proof. Top-responsiveness follows from Lemma B.1; informativeness from Lemma B.2. The PAC-stability guarantee is therefore an immediate corollary of Theorem 3.4 in (Sliwinski & Zick, 2017). □

B.2 COALITION-LEVEL VALUATION (FOR SAMPLING)

Algorithm **??** needs a scalar value for any sampled coalition S. We use the symmetric extension

$$\Phi(S) \;=\; \frac{1}{|S|} \sum_{i \in S} u_i(S) \;=\; \frac{1}{|S|} \sum_{i \in S} \sum_{j \in \texttt{Top}_k(i,S)} \phi_{ij}.$$

Intuitively, $\Phi(S)$ averages how strongly each member is bonded to its k preferred partners within S. Plugging ϕ_{OCA} or ϕ_{PAS} in place of ϕ_{ij} yields the concrete scores used in our experiments. "Reservoir" sampling in line 4 of Algorithm **??** draws m subsets S with probability proportional to $\Phi(S)$, thereby prioritising high-synergy groups.

[3] Adding a partner can only increase the set of k best partners or leave it unchanged.

Algorithm 1 PAC Top-Cover for Top-k Responsive Games (neurons)

Require: $\phi \in \mathbb{R}^{n\times n}$ ▷ pairwise affinity; $\phi_{ii} = 0$
$k \in \mathbb{N}$ ▷ top-k choice size
m, ω ▷ reservoir size, per-round samples
MINK, MAXK ▷ sampled coalition sizes
(ε, δ) ▷ PAC guidance for m, ω

$R \leftarrow \{1, \dots, n\}, \quad \pi \leftarrow \emptyset$
$S \leftarrow \textsc{SampleCoalitions}(R, m, \textsc{MinK}, \textsc{MaxK})$ ▷ reservoir

Definition (top-k utility and choice in a coalition).
For $i \in T$, let $P_i(T) = T \setminus \{i\}$. Let $\textsc{TopK}_i(T)$ be the k indices in $P_i(T)$ with largest ϕ_{ij} (ties broken by smaller index); if $|P_i(T)| < k$, take all.
Define $u_i^k(T) \triangleq \frac{1}{|\textsc{TopK}_i(T)|} \sum_{j\in\textsc{TopK}_i(T)} \phi_{ij}$.

while $R \neq \emptyset$ **do**
  $S_{\text{round}} \leftarrow$ first ω sets in S that satisfy $T \subseteq R$; remove them from S
  if $|S_{\text{round}}| < \omega$ **then** ▷ refresh if reservoir depleted
    $S \leftarrow S \cup \textsc{SampleCoalitions}(R, m, \textsc{MinK}, \textsc{MaxK})$
  end if
  for all $i \in R$ **do**
    $\mathcal{T}_i \leftarrow \{T \in S_{\text{round}} : i \in T\}$
    if $\mathcal{T}_i = \emptyset$ **then**
      $B_i \leftarrow \{i\}$ ▷ degenerate self-loop
    else
      $T_i^\star \leftarrow \arg\max_{T\in\mathcal{T}_i} u_i^k(T)$ ▷ deterministic tie-break by T's lexicographic index list
      $B_i \leftarrow \textsc{TopK}_i(T_i^\star)$ ▷ top-k choice set of i in $T_i^\star$
    end if
  end for
  Build digraph $G = (R, E)$ with edges $(i \to j)$ for all $j \in B_i$ (and optional $(i \to i)$ self-loops)
  Let $\mathcal{C} \leftarrow$ the set of sink strongly connected components of G
  (closure check) Keep only $X \in \mathcal{C}$ such that $\forall i \in X : \ B_i \subseteq X$
  Choose $X \in \mathcal{C}$ (e.g., smallest by size then lexicographic) ▷ any sink closed SCC is valid
  $\pi \leftarrow \pi \cup \{X\}; \quad R \leftarrow R \setminus X$
end while
return π

C PAC TOP COVER ALGORITHM

D INFORMATION RETRIEVAL PRELIMINARIES

Information Retrieval (IR) involves retrieving documents that are likely to be relevant to a user's information need, typically represented as a query. A fundamental IR task is to return a ranked list of documents in descending order of (estimated) relevance. The quality of this ranking directly impacts the user experience in search engines, recommendation systems, and question answering applications.

D.1 RELEVANCE MODEL

Dense/Neural Re-ranker is a language model (like RankLLaMa (Ma et al., 2024a)) which takes a query and text as input and produces a relevance score based on the similarity of the query to the provided text.

Relevance Modeling vs Classification Classification and relevance modeling are related but distinct approaches in information retrieval (IR). The term relevance model (Lavrenko & Croft, 2001) refers to a mechanism for estimating the likelihood of observing a particular word in documents that are relevant to a given information need or query, whereas classification assigns documents to predefined categories, such as relevant or non-relevant.

Ranking Evaluation with NDCG

In information retrieval, one commonly used metric to evaluate the effectiveness of ranking models is the Normalized Discounted Cumulative Gain (NDCG). NDCG assesses the quality of a ranked list by measuring the gain (or relevance) of documents based on their position in the list, giving higher weight to relevant documents that appear earlier. Formally, the Discounted Cumulative Gain (DCG) is computed as:

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

where rel_i is the graded relevance of the document at position i. The NDCG is then computed by normalizing DCG by the ideal DCG (IDCG), which is the DCG for the optimal ranking:

$$\text{NDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}}$$

NDCG scores range from 0 to 1, with 1 indicating a perfect ranking. In the DL19 dataset, each query-document pair is labeled with a relevance grade based on human annotations. These annotations are used to compute the NDCG score for a re-ranked list of documents, allowing us to quantify the effectiveness of our rerankers in retrieving the most relevant content at the top of the list.

In our work, we use **RankLLaMA**, a LLaMA-based reranking model trained to predict the relevance of a document given a query. The model takes as input a formatted string:

```
"query: {query}, passage: {passage}"
```

and outputs a score between 0 and 1, indicating the estimated relevance. We follow the training procedure described in the RankLLaMA paper (Ma et al., 2024a).

D.2 COVERED QUERY TERM RATIO (CQTR)

Covered Query Term Ratio (CQTR) is a lexical feature that measures the proportion of unique query terms found in the document(Qin & Liu, 2013a). Formally:

$$\text{CQTR} = \frac{|\text{Query Terms} \cap \text{Document Terms}|}{|\text{Query Terms}|}$$

D.3 Mean Term Frequency per Document Length (MTF/L)

Mean Term Frequency per Document Length (MTF/L) captures the average frequency of query terms normalized by the document length(Qin & Liu, 2013a). It is computed as:

$$\text{MTF/DL} = \frac{\sum_{t \in Q} \text{TF}_t(D)}{\text{Length}(D)}$$

To simplify interpretability tasks (by trying to restrict polysemanticity(Elhage et al., 2022)), we fine-tuned models on CQTR and MTF/L prediction tasks, with the same input structure as defined above. We do not claim that these two features are the most important for determining relevance; rather, they are easily understood signals that prior work has shown to be implicitly present in neural models (Chowdhury et al., 2025).

D.4 Datasets

Datasets:

- **MS MARCO**: A large-scale dataset consisting of real anonymized web search queries paired with relevant passages. It is a standard benchmark for training and evaluating re-ranking models. In our fine-tuning, we used a modified version of this dataset called MS MARCO Augmented (Tevatron, 2024) (Ma et al., 2025), which provides hard negatives from both CoconDenser(Gao et al., 2021) and BM25. [4]
- **DL-19 (TREC Deep Learning Track 2019)**: Contains high-quality relevance annotations for a subset of queries, commonly used for zero-shot and fine-tuned re-ranker evaluation. Craswell et al. provide more information and an overview of this dataset (Craswell et al., 2020).

[4] More details at https://microsoft.github.io/msmarco/

E FEATURE/LAYER CHOICE

Previous interpretability studies have been conducted on dense re-rankers, where Chowdhury et al. found that using linear probing, several traditional IR features show a high likelihood of being present in the forward pass activations of a dense re-ranker model(Chowdhury et al., 2025). Further behavioral analysis by Nijasure et al. observed that large language models (LLMs) tend to learn relevance-related features primarily in MLP layers 5 to 14 of re-ranker architectures(Nijasure et al., 2025).

Motivated by these insights, we focused our probing and editing experiments on this layer range (5–14) of Re-Ranker models. Figure 1 supports this choice: it shows R^2 scores for predicting MSLR features across all layers of the RankLLaMA-7b model using linear probing. Features like *covered query term number*, *covered query term ratio*, *mean of stream length normalized term frequency*, and *variance of* $tf \cdot idf$ exhibit increasing prominence from the lower to mid layers. This trend might indicate that these layers are key to encoding relevance-related signals.

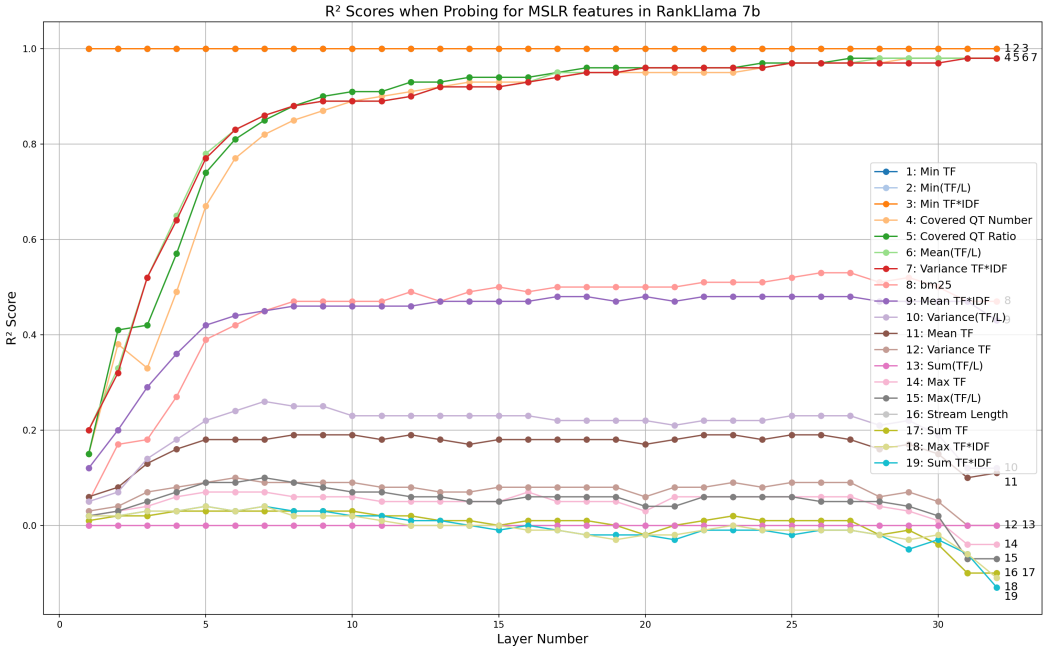

Figure 1: Probing for statistical features from the MSLR dataset in RankLlama2-7b model. Here QT stands for Query Term, TF stands for Term Frequency and $\cdot/L$ stands for length normalized. The graph lines indicate the presence of a particular feature along the layers of the LLM. Certain features like $Min\ TF * IDF$ show consistent presence across the layers. Other features like $Covered\ QT\ Number$, $Covered\ QT\ Ratio$, $Mean(TF/L)$ and $Variance\ TF * IDF$ show increasing prominence from the first layer to the last, ultimately playing an important role in making ranking decisions. Other MSLR features like $Sum(TF/L)$, $Max(TF/L)$, and $Sum\ TF * IDF$ show negative correlation with RankLlama decision making(Chowdhury et al., 2025).

F LLM PERFORMANCE EVALUATION

We used LoRA (rank 8) fine-tuning on MLP modules alone for all models described in this section. We had access to four A100 GPUs, depending on availability. We used DeepSpeed's Stage 0 configuration with the AdamW optimizer for fine-tuning all these models.

Two of the LLMs used in our experiments were fine-tuned on the MS MARCO dataset for 0.3 epochs using Mean Squared Error (MSE) as the loss function. The models were trained to predict statistical IR signals such as the Covered Query Term Ratio (CQTR) and the mean term frequency normalized by passage length (mean(TF/L)). Following this fine-tuning, the models were evaluated on a sampled subset of the DL19 dataset. This evaluation set comprised 43 queries, each associated with 10 documents sampled from a larger candidate pool of 200 documents per query, retrieved using the ReplLLaMA retriever. This setup was designed to assess the models' ability to learn and generalize statistical IR features relevant to document ranking. Table 5 summarizes the finetuning results of the LLMs.

For fine-tuning the re-rankers, we used the code provided in the Tevatron repository (Ma et al., 2024b). For more details, refer to the paper by (Ma et al., 2024a). Evaluation was conducted on the full DL19 dataset, with document ranking based on the top 200 passages retrieved via the ReplLLaMA retriever. Results for finetuned re-rankers is presented in the table 4.

Base LLM	**Target Feature**	**Base NDCG@10**	**NDCG@10 (Finetuned)**
LLaMA3(Dubey et al., 2024)	Re-Ranking	0.18	0.7497
Pythia(Biderman et al., 2023)	Re-Ranking	0.18	0.7521
Mistral(Jiang et al., 2023)	Re-Ranking	0.18	0.7570

Table 4: NDCG@10 evaluation on DL19 dataset, showing baseline vs post-finetuning performance. All models were fine-tuned on MS MARCO for 1 epoch.

Base LLM	**Function**	**MSE (Start)**	**MSE (Finetuned, 0.3 epoch)**
LLaMA3(Dubey et al., 2024)	CQTR	3.88	0.52
Pythia(Biderman et al., 2023)	CQTR	1.84	0.05
Mistral(Jiang et al., 2023)	CQTR	36.92	10.94
LLaMA3(Dubey et al., 2024)	mean(TF/L)	5.06	4.49
Pythia(Biderman et al., 2023)	mean(TF/L)	2.24	0.00
Mistral(Jiang et al., 2023)	mean(TF/L)	38.32	22.23

Table 5: MSE before and after finetuning (0.3 epochs) for CQTR and mean(TF/L) prediction tasks on the sampled DL19 dataset.

Table 6: Coalition Synergy (↑) measured via Pairwise and ratio: mean ±95% CI across three seeds.

Task / Algorithm	LLaMA-3.1		Mistral		Pythia	
	Pairwise	Ratio	Pairwise	Ratio	Pairwise	Ratio
Covered Query Term Ratio						
Random	0.01±0.05	0.49±0.04	−0.02±0.06	0.53±0.05	0.00±0.05	0.50±0.04
K-means	−0.23±0.03	0.32±0.03	−0.20±0.04	0.36±0.03	−0.18±0.03	0.37±0.03
Hier. clustering	−0.11±0.04	0.41±0.03	−0.13±0.04	0.43±0.03	−0.17±0.04	0.40±0.03
Hedonic (OCA)	0.08±0.01	0.74±0.02	0.10±0.01	0.71±0.02	0.06±0.01	0.78±0.02
Hedonic (PAS)	**0.12**±0.005	**0.86**±0.01	**0.13**±0.005	**0.84**±0.01	**0.15**±0.006	**0.89**±0.01
Mean of Normalized Term Frequency						
Random	0.02±0.05	0.41±0.04	−0.01±0.05	0.53±0.05	0.00±0.05	0.50±0.04
K-means	−0.22±0.03	0.34±0.03	−0.21±0.03	0.35±0.03	−0.16±0.03	0.31±0.03
Hier. clustering	−0.08±0.04	0.43±0.03	−0.08±0.04	0.43±0.03	−0.15±0.04	0.39±0.03
Hedonic (OCA)	0.01±0.01	0.72±0.02	0.04±0.01	0.77±0.02	0.03±0.01	0.74±0.02
Hedonic (PAS)	**0.09**±0.006	**0.85**±0.01	**0.14**±0.006	**0.82**±0.01	**0.16**±0.007	**0.89**±0.01
Relevance						
Random	0.01±0.05	0.42±0.04	−0.02±0.05	0.44±0.04	0.03±0.05	0.49±0.04
K-means	−0.13±0.03	0.33±0.03	−0.14±0.03	0.36±0.03	−0.19±0.03	0.38±0.03
Hier. clustering	−0.09±0.04	0.42±0.03	−0.12±0.04	0.44±0.03	−0.08±0.04	0.44±0.03
Hedonic (OCA)	0.05±0.01	0.77±0.02	0.05±0.01	0.75±0.02	0.04±0.01	0.73±0.02
Hedonic (PAS)	**0.11**±0.005	**0.81**±0.01	**0.13**±0.005	**0.87**±0.01	**0.14**±0.006	**0.86**±0.01

G Intrinsic Coalition Evaluation

Synergy Metrics. Let x be an input sampled from the task distribution $\mathcal{D}$ and $\ell(x) \in \mathbb{R}$ the *layer-local logit* (defined in §3.2) with all neurons active. For any neuron set S we denote by $\ell_{-S}(x)$ the same forward pass after zeroing the activations of every $k \in S$ *only* inside the LoRA-adapted MLPs. We define the *marginal contribution* of a single neuron as $\psi(i) = \mathbb{E}_{x\sim\mathcal{D}}[\ell(x) - \ell_{-\{i\}}(x)]$, and the *pairwise interaction (synergy)* of two neurons as $\psi(i,j) = \mathbb{E}_{x\sim\mathcal{D}}[\ell(x) - \ell_{-\{i\}}(x) - \ell_{-\{j\}}(x) + \ell_{-\{i,j\}}(x)]$. A positive $\psi(i,j)$ means that removing *both* neurons harms the logit more than the sum of their individual removals (synergy), while a negative value indicates redundancy. For a coalition $C \subseteq \{1,\dots,n\}$ we report two size-agnostic aggregates: $\text{Pair}(C) = \frac{1}{|C|(|C|-1)} \sum_{\substack{i,j\in C \\ i\neq j}} \psi(i,j)$ and $\text{Ratio}(C) = \frac{\sum_{i\neq j\in C} \psi(i,j)}{\sum_{i\in C} \psi(i)}$. *Pairwise Synergy* is the mean interaction strength across all ordered neuron pairs, fully normalized for coalition size, while *Ratio Synergy* compares the *extra* value created by pairwise cooperation (numerator) to the value explained by separate single-neuron effects (denominator). A ratio near 1 or greater (*super-additivity*) indicates that the coalition's joint influence exceeds the sum of its parts, whereas a ratio near 0 (or negative) signals antagonistic or redundant behavior.

Intrinsic Evaluation Results. Regarding synergy quality (Table 6), the two hedonic variants strictly dominate all baselines across all three backbones and all three MS-MARCO objectives: *Hedonic-PAS* attains the best Pairwise *and* Ratio score in 26 out of 27 model-metric cells, while *Hedonic-OCA* follows as a close second. Relative to spherical k-means, the average margin is +0.29 Pairwise and +0.49 Ratio, indicating that activation similarity alone is a poor proxy for *functional* cooperation. Random and hierarchical clusterings even dip into negative Pairwise values (sub-additivity) and hover near the additive boundary on the Ratio metric, underscoring the value of an explicit game-theoretic objective. Confidence intervals (95%, $df = 2$) never overlap between Hedonic-PAS and the best baseline, with paired t-tests yielding $p < 0.01$ for every layer. K-means/HAC produce fairly uniform sizes (20-45 neurons per cluster), whereas hedonic output follows a heavy-tailed Zipf-like law: each layer contains a single "macro" coalition (> 150 neurons), ~ 100 coalitions of size 2, and approximately 500 clusters with $|C| > 1$ covering $\sim 14,000$ neurons. In most settings, the top cover algorithm converges with reservoir size $m \leq 120,000$ and number of samples per iteration $\omega \leq 32,000$.

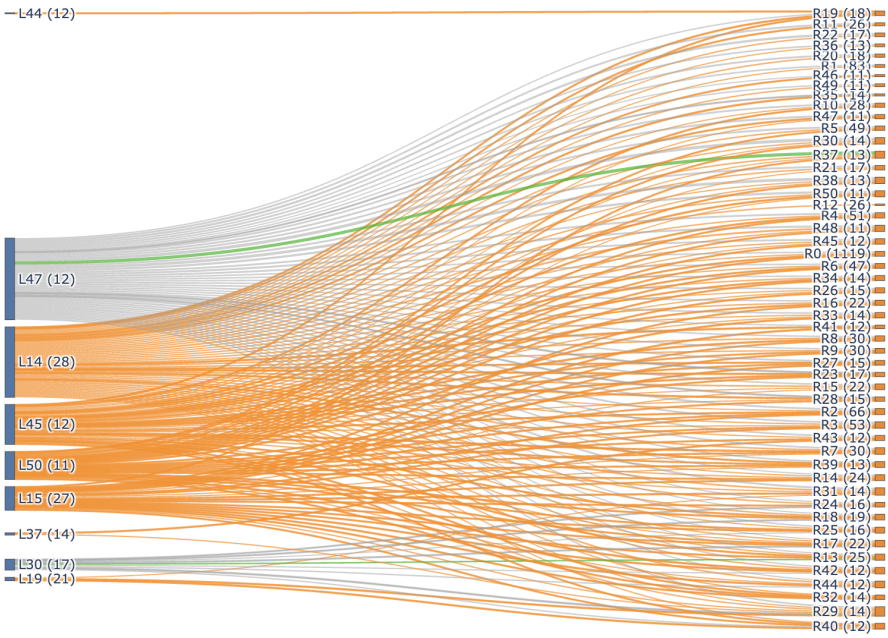

Figure 2: Coalition flow across depth for MISTRAL-7B fine-tuned on the *relevance modelling* task. The figure depicts flow from layers $7 \rightarrow 8$, with orange depicting *split*, green depicting *persist* and grey depicting *vanish* of coalitions.

H COALITION FLOW EXAMPLE