Emergent Semantic Role Understanding in Language Models

Carla Griffiths
University College London

Mirco Musolesi
University College London
University of Bologna

Abstract

Understanding how linguistic structure emerges in language models is central to interpreting what these systems learn from data and how much supervision they truly require. In particular, semantic role understanding ("who did what to whom") is a core component of meaning representation, yet it remains unclear whether it arises from pre-training alone or depends on task-specific fine-tuning. We study whether semantic role understanding emerges during language model pre-training or requires task-specific fine-tuning. We freeze decoder-only transformers and train linear probes to extract semantic roles, using performance to infer whether role information is already encoded in pre-training or learned during adaptation. Across model scales, we find that frozen representations contain substantial semantic role information, with performance improving but not fully matching fine-tuned models. This indicates partial but incomplete emergence from pre-training alone. We show that semantic role structure emerges from language modeling objectives, but its internal implementation shifts toward more distributed representations as model scale increases.

1 Introduction

Understanding the emergence of linguistic structure in Large Language Models (LLMs) is essential for characterizing what they learn from data and the extent to which supervision is required. In particular, *semantic role understanding* ("who did what to whom") Gildea and Jurafsky [2002] is a core component of meaning representation, yet it remains unclear whether it arises from pre-training alone or depends on task-specific fine-tuning. Indeed, LLMs trained on next-token prediction have demonstrated capabilities that arguably extend beyond simple text completion [Brown et al., 2020, Radford et al., 2019]. Building on the transformer architecture [Vaswani et al., 2017] and pre-training paradigms pioneered by BERT [Devlin et al., 2019] and ELMo [Peters et al., 2018a], modern LLMs learn rich contextual representations that generalize across tasks [Howard and Ruder, 2018, Ruder et al., 2019]. This success has prompted extensive investigation into what linguistic knowledge these models acquire [Rogers et al., 2020, Jawahar et al., 2019, Liu et al., 2019].

In their purest form, language models are designed to generate coherent text that can be understood by another human, reflecting a fundamental aspect of human communication. If text generation is an inherently human characteristic, it is natural to ask whether semantic roles emerge in machine-generated text in ways comparable to those observed in human communication. Semantic roles provide a simple way to ask whether the "who did what to whom" structure is encoded in a network's weights, or only surface statistics.

Our central hypothesis is that *semantic role understanding emerges from pre-training alone*, even at small model scales, and that this emergence can be detected and characterized through probing and targeted ablations. Semantic role labeling (SRL) is the task of identifying the predicate-argument structure of sentences, determining "who did what to whom, when, where, and how" [Palmer et al.,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

2010]. For example, given the sentence "The dog took a long walk in the park," QA-SRL generates questions like "Who took a walk?" (Agent: the dog), "What did someone take?" (Patient: a long walk), and "Where did someone take a walk?" (Location: the park). This formulation is particularly suitable for probing language models, as it requires no explicit linguistic annotation schema and naturally maps to extractive question answering. We measure QA-SRL answer-span prediction, not full semantic role understanding. QA-SRL span prediction is a small but important building block for studying emergence[1] in this setting. We chose to study this problem because of its foundational nature and its importance. Our goal is to focus on a simple, highly decodable task rather than a larger environment-based experiment, which may be more difficult to interpret.

How can we tell if a model learned semantic roles during pre-training rather than during task-specific training? We freeze the pre-trained model and train only a simple linear layer to extract answers. If this frozen model performs well, semantic role information was encoded during pre-training; if the frozen model performs poorly, the capability must be learned during fine-tuning. In particular, our experimental design addresses three core questions:

1. *Does emergence of semantic role understanding occur?* Do frozen model representations contain sufficient information for semantic role identification?
2. *Is emergence of semantic role understanding scale-dependent?* At what model scale (if any) does the gap between frozen-probe and full-fine-tuning performance narrow significantly?
3. *How is semantic role information organized?* Can we identify how semantic roles are encoded inside the model and the intrinsic neural mechanisms? Do these representations form a clear, interpretable structure?

Beyond linear probing, we perform mechanistic analysis to understand how semantic roles are represented. Using PCA and t-SNE [van der Maaten and Hinton, 2008] visualization of layer activations, following similar approaches in LLM interpretability [Gurnee and Tegmark, 2024], we find that semantic roles form distinct clusters in representation space (Figure 6). We identify individual neurons in the feed-forward layers that selectively activate for specific semantic roles (e.g., "*Agent* neurons", "*Location* neurons") and causally validate their importance through targeted ablation experiments. These role-selective neurons exhibit functional co-activation patterns, with within-role correlations exceeding those across roles.

We train decoder-only transformer models from scratch on WikiText-103 [Merity et al., 2016], spanning four scales from 0.4M to 57M transformer parameters. We evaluate using QA-SRL [He et al., 2015], a task that reformulates semantic role labeling as question answering (e.g., asking "Who took a walk?" to identify the Agent role). Our results characterize the emergence threshold for semantic role understanding and contribute to the broader understanding of what linguistic capabilities arise from language modeling objectives alone.

This work makes the following contributions:

- We present a systematic methodology for testing emergence of semantic role understanding using linear probing on frozen LLM representations. We provide empirical evidence characterizing the relationship between model scale and the frozen-probe vs. full-fine-tuning performance gap on QA-SRL.
- We introduce a temporal analysis framework using Centered Kernel Alignment (CKA) to track how representations evolve during fine-tuning relative to the pre-trained (null) model.
- We identify role-selective neurons through PCA/t-SNE visualization and neuron ablation, discovering that semantic roles cluster in representation space.
- We conduct role-conditioned neuron ablations revealing scale-dependent neural reorganization: *Agent*-selective neurons transition from causally important in small models to interfering in large models, while *Time*-selective neurons remain causal across larger model scales.

Code and trained models with checkpoint weights are available at the following URL: https://github.com/carlacodes/emergent-semantic-roles.

[1]We use "emergence" to refer to the de novo learning of relationships between grammatical structures without supervised fine-tuning or explicit instruction.

2 Related Work

Emergence in Language Models. *Emergent abilities* in language models refer to capabilities that appear suddenly as models scale [Wei et al., 2022]. Scaling laws predict smooth improvements in perplexity [Kaplan et al., 2020, Hoffmann et al., 2022], but task performance can exhibit sharp transitions. However, recent work questions whether emergence is a genuine phase transition or a measurement artifact: Schaeffer et al. [2023] argue that nonlinear metrics like exact match create the appearance of sudden capability acquisition, while Snell et al. [2024] show that fine-tuning can unlock capabilities present but inaccessible in smaller models. These findings suggest that "emergence" may reflect our evaluation methods rather than underlying model organization. Our work addresses this limitation by using continuous F1 metrics and linear probing, which provide a more fine-grained view of capability development. Rather than asking *when* semantic role understanding appears, we characterize *how much* is present at each scale, distinguishing between information encoded during pre-training versus information created during fine-tuning.

Probing Classifiers. Probing classifiers analyze what information is encoded in neural representations [Belinkov, 2022]. The BERTology literature [Rogers et al., 2020, Jawahar et al., 2019, Clark et al., 2019, Liu et al., 2019, Vulić et al., 2020, Peters et al., 2018b] has extensively applied probing, finding that lower layers encode syntax while higher layers capture semantics. Linear probes are preferred to constrain capacity [Hewitt and Liang, 2019]. Our methodology extends probing by comparing frozen performance to full fine-tuning, measuring how much information is already present versus must be learned.

Semantic Role Labeling. SRL has evolved from feature-based classifiers [Gildea and Jurafsky, 2002, Pradhan et al., 2005] to neural approaches [Zhou and Xu, 2015, He et al., 2017, Marcheggiani and Titov, 2017, Roth and Lapata, 2016, Strubell et al., 2018], with pre-trained models providing strong features [Shi and Lin, 2019]. We use QA-SRL [He et al., 2015, FitzGerald et al., 2018], which reformulates SRL as question answering without requiring explicit PropBank schemas, making it suitable for probing emergence from language exposure alone.

Linguistic Structure in LLMs. Structural probes find syntactic tree structure in LLM representations [Hewitt and Manning, 2019], while layer-wise analysis shows that lower layers encode surface features and higher layers capture abstract semantics [Tenney et al., 2019, Peters et al., 2018b, Liu et al., 2019]. Building on static [Mikolov et al., 2013, Pennington et al., 2014] and contextualized embeddings [Peters et al., 2018a, Devlin et al., 2019], our work examines semantic *role* structure specifically, using ablations to establish causal relationships between neurons and semantic functions.

Mechanistic Interpretability. Recent work reverse-engineers neural network algorithms: feed-forward layers as key-value memories [Geva et al., 2021], circuit-level task analysis [Wang et al., 2023, Conmy et al., 2023, Olsson et al., 2022], and sparse autoencoders for monosemantic features [Elhage et al., 2022, Bricken et al., 2023, Cunningham et al., 2024, Templeton et al., 2024]. Our work differs by training models from scratch for controlled scale manipulation, focusing on QA-SRL as a well-defined semantic task, and combining probing with neuron ablation to test causal necessity.

3 Methods

3.1 Model Architecture

We train decoder-only transformer language models following the GPT-2 & GPT-3 architecture [Radford et al., 2019]. All models use learned positional embeddings, pre-layer normalization, GELU activations, and a feed-forward expansion factor of 4. We use the `bert-base-uncased` tokenizer ($\sim$30K vocabulary), held fixed across all four scales following standard practice in scaling-law studies [Kaplan et al., 2020, Hoffmann et al., 2022]. Scaling vocabulary with model size would confound transformer capacity with tokenization granularity; under our design, the same input distribution is presented to all four transformers, so the only varying quantity is the transformer's depth and width. We tie token embedding weights to the LM head during pre-training. We train four model configurations spanning approximately three orders of magnitude in parameter count (Table 1).

Table 1: Model configurations and parameter counts. Total parameters include embeddings (token + positional); transformer parameters exclude embeddings to enable comparison across vocabulary sizes.

Config	Layers	Hidden	Heads	Transformer	Total
Tiny	2	128	2	0.4M	4.4M
Small	4	256	4	3.2M	11.1M
Base	6	512	8	18.9M	34.8M
Medium	8	768	12	56.7M	80.5M

3.2 Pre-training

All models are pre-trained on WikiText-103 [Merity et al., 2016] using the standard causal language modeling objective. WikiText-103 contains approximately 103 million tokens of Wikipedia articles, providing exposure to diverse factual content and linguistic structures without explicit QA formatting.

The pre-training objective is next-token prediction with cross-entropy loss:

$$\mathcal{L}_{\text{pretrain}} = -\sum_{t=1}^{T} \log P(x_t | x_{<t}; \theta) \tag{1}$$

where x_t is the token at position t and θ represents model parameters.

3.3 QA-SRL Fine-tuning Task

After pre-training, models are fine-tuned (or probed) on QA-SRL [He et al., 2015]. Each example consists of a sentence, a predicate within that sentence, a question about that predicate's semantic role, and an answer span. During preprocessing, we skip examples where the answer text appears multiple times in the context (ambiguous spans), filtering approximately 9% of examples. The model receives the concatenated input:

`[question] [SEP] [sentence]`

and must predict start and end positions for the answer span within the sentence. Critically, we maintain causal attention during fine-tuning (matching pre-training), ensuring that any performance gap between frozen and fine-tuned models reflects differences in learned representations rather than attention pattern mismatch. With the question preceding the sentence, the model can attend to the full question when processing each sentence token.

We add a QA head consisting of two linear projections from the hidden dimension to scalar logits for start and end positions:

$$\mathbf{s} = \mathbf{H}\mathbf{w}_{\text{start}} \tag{2}$$

$$\mathbf{e} = \mathbf{H}\mathbf{w}_{\text{end}} \tag{3}$$

where $\mathbf{H} \in \mathbb{R}^{L \times d}$ is the sequence of hidden states and $\mathbf{w}_{\text{start}}, \mathbf{w}_{\text{end}} \in \mathbb{R}^d$ are learned projection vectors.

3.4 Linear Probing Protocol

Overview. Our protocol distinguishes two related claims that prior probing work has often conflated. Information is *decodable* when a linear probe extracts it from a frozen pre-trained model, establishing the information is present somewhere in the model's parameters and inputs together. It is *emergent* from pre-training when the information lives in the pre-trained transformer weights, separately from trainable embeddings or fine-tuned layers, evidenced by a gap over an architecture-matched random baseline and corroborated by per-layer probes on pure pre-trained representations with zero embedding adaptation (Section 4.4). Thus, we are able to test emergence under controlled conditions rather than general decodability. Note that the token embedding is a position-independent lookup table (identical outputs for "the dog bit the child" and "the child bit the dog"), so role resolution under the frozen probe must come from the frozen transformer's cross-position attention rather than from embedding adaptation alone. We use two fine-tuning regimes described below.

Full Fine-tuning and Frozen Linear Probe. In *full fine-tuning*, all model parameters (transformer layers and QA head) are updated during QA-SRL training. This establishes an upper bound on task performance for each model scale. In the case of the *frozen linear probe*, all transformer block parameters and positional embeddings are frozen; only the token embeddings and QA head ($\mathbf{w}_{\text{start}}, \mathbf{w}_{\text{end}}$) are trained. This tests whether the answer span information is encoded in the frozen transformer representations. We deliberately keep the token embedding matrix trainable rather than frozen: the input format `[question] [SEP] [sentence]` relies on `[SEP]` as a structural separator, and `[SEP]` is a BERT-tokenizer artifact that was never seen in a separator role during causal language-model pre-training, so a fully frozen embedding matrix would prevent the model from parsing the input format and confound representation quality with input-format adaptation rather than test it. Embedding updates can still propagate through the frozen layers, creating a residual "adaptation effect" that we analyze separately via per-layer probes with zero embedding adaptation (Section 4.4) and a stricter frozen-embedding (SEP-only) ablation in Appendix I, where only `[SEP]` and the QA head update. The gap between full fine-tuning and frozen probe performance quantifies how much semantic role information must be learned during fine-tuning versus how much is already present from pre-training.

Random Baseline. To establish a performance floor, we also evaluate a *random baseline*: a randomly initialized (untrained) model whose token embeddings and QA head are trained on QA-SRL while the transformer blocks remain at random initialization. This is a capacity-matched comparator for the frozen probe (same trainable parameters, only the source of the transformer weights differs), which isolates the contribution of the pre-trained transformer from any contribution of trainable embeddings.

Normalized Emergence Score. We define a normalized emergence score that accounts for the random baseline floor:

$$\text{Emergence}_{\text{norm}} = \frac{F1_{\text{frozen}} - F1_{\text{random}}}{F1_{\text{full}} - F1_{\text{random}}}. \tag{4}$$

A score near 1.0 indicates strong emergence (frozen probe captures most of what full fine-tuning achieves above chance), while a score near 0.0 indicates the frozen probe performs no better than random initialization. We also report the raw gap $\Delta_{\text{emergence}} = F1_{\text{full}} - F1_{\text{frozen}}$ (while less rigorous, the raw score is more intuitive). Emergence scores should be read as information availability rather than predictions of generation-mode performance.

3.5 Temporal CKA Analysis

To understand how representations evolve during fine-tuning, we employ Centered Kernel Alignment (CKA) [Kornblith et al., 2019] to measure representational similarity across training checkpoints. For activation matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times d}$, linear CKA is:

$$\text{CKA}(\mathbf{X}, \mathbf{Y}) = \frac{\|\mathbf{Y}^\top \mathbf{X}\|_F^2}{\|\mathbf{X}^\top \mathbf{X}\|_F \|\mathbf{Y}^\top \mathbf{Y}\|_F}. \tag{5}$$

We save checkpoints at each epoch and compute CKA between each fine-tuning checkpoint and the pre-trained (*null*) model. A decrease in CKA_{null} indicates representational drift from pre-training during fine-tuning.

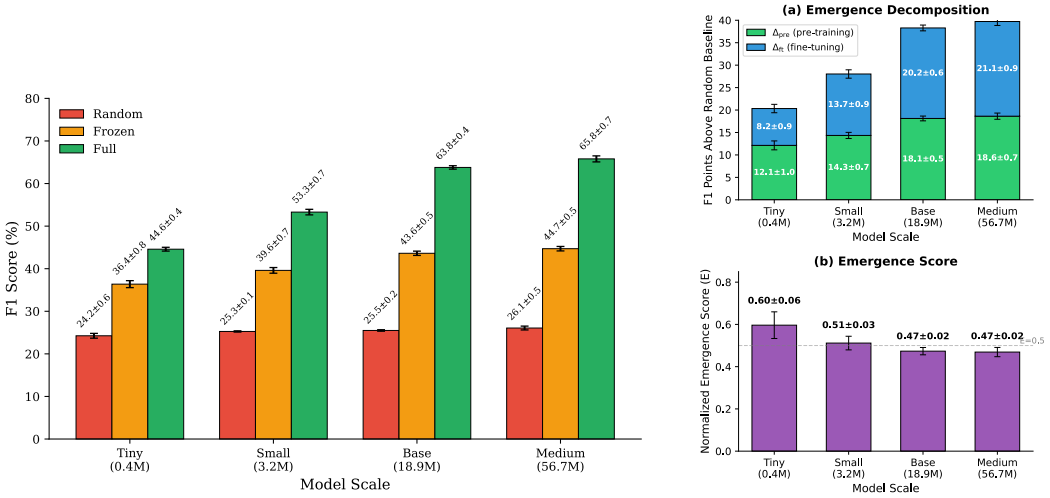
4 Results

4.1 Overview

We present results across four model scales: Tiny (0.4M), Small (3.2M), Base (18.9M), and Medium (56.7M transformer parameters). We report the F1 Score on the QA-SRL validation set, defined as the token-level overlap between predicted and gold answer spans, computed as the harmonic mean of precision and recall. In Table 2, each run reports its best validation F1 over fine-tuning epochs. In the evaluation, we apply context masking to constrain predictions to the sentence region only, preventing the model from predicting spans within the question or special tokens. Table 2 summarizes performance across three experimental conditions for each model size: random baseline (untrained LLM with trained QA head), frozen linear probe (pre-trained LLM frozen, only QA head trained), and full fine-tuning (all parameters updated).

Table 2: Results on QA-SRL semantic role labeling across model scales. All models show positive emergence, with frozen pre-trained representations encoding semantic role information accessible to linear probes. F1 (%) shown as mean±std over 5 random seeds; emergence score $E = \Delta_{\text{pre}}/(\Delta_{\text{pre}} + \Delta_{\text{ft}})$.

	Tiny (0.4M)	Small (3.2M)	Base (18.9M)	Medium (56.7M)
Random baseline	24.2±0.6	25.3±0.1	25.5±0.2	26.1±0.5
Frozen (linear probe)	36.4±0.8	39.6±0.7	43.6±0.5	44.7±0.5
Full fine-tuning	44.6±0.4	53.3±0.7	63.8±0.4	65.8±0.7
Emergence E	0.60±0.06	0.51±0.03	0.47±0.02	0.47±0.02



(a) F1 across conditions and scales.

(b) Emergence decomposition.

Figure 1: Frozen probes extract semantic role information without any model adaptation. (a) We compare three conditions: *Random baseline* (untrained model + QA head only), *Frozen probe* (pre-trained model frozen, only QA head trained), and *Full fine-tuning* (all parameters updated). The frozen probe substantially outperforms random at all scales. (b) Decomposing performance into $\Delta_{\text{pre}} = F1_{\text{frozen}} - F1_{\text{random}}$ (pre-training above chance) and $\Delta_{\text{ft}} = F1_{\text{full}} - F1_{\text{frozen}}$ (fine-tuning's additional contribution): pre-training contributes 12–19 F1 points across scales while fine-tuning's contribution grows with scale (8–21 points). The normalized emergence score $E = \Delta_{\text{pre}}/(\Delta_{\text{pre}} + \Delta_{\text{ft}})$ decreases with scale because fine-tuning becomes increasingly effective relative to a relatively constant pre-training contribution. Values: mean ± std over 5 random seeds.

4.2 Analysis of the Emergence of Semantic Role Understanding

All models show positive emergence, with frozen probes consistently outperforming random baselines (Figure 1b). To understand what drives the emergence score, we decompose it into two components:

- $\Delta_{\text{pre}} = F1_{\text{frozen}} - F1_{\text{random}}$: pre-training's contribution above chance;
- $\Delta_{\text{ft}} = F1_{\text{full}} - F1_{\text{frozen}}$: fine-tuning's additional contribution.

The normalized emergence score is defined as $E = \Delta_{\text{pre}}/(\Delta_{\text{pre}} + \Delta_{\text{ft}})$. Examining these components reveals that pre-training's contribution (Δ_{pre}) ranges from 12.1 to 18.6 F1 points (12.1±1.0, 14.3±0.7, 18.1±0.5, 18.6±0.7 for Tiny through Medium), showing consistent gains across all scales. In contrast, fine-tuning's contribution (Δ_{ft}) grows substantially with scale (8.2±0.9, 13.7±0.9, 20.2±0.6, 21.1±0.9 points). The normalized emergence score decreases with scale because fine-tuning becomes increasingly effective at larger scales while pre-training's contribution stays roughly constant. The score thus reflects the fraction of task performance that comes "for free" from pre-training, and this fraction shrinks as fine-tuning scales up.

These results show that semantic role information is decodable from frozen pre-trained representations across all scales, with a consistent 12–19 F1 gap above the architecture-matched random baseline. The main frozen probe (Table 2) still trains token embeddings; Appendix I (Table 14) freezes almost all embeddings except [SEP]. The Tiny/Small/Base/Medium F1 scores are: pre-trained with embeddings trainable: 36.4/39.6/43.6/44.7 vs. SEP-only: 23.9/27.9/30.5/31.6; random with embeddings trainable: 24.2/25.3/25.5/26.1 vs. SEP-only: 24.4/27.4/27.8/28.0. Thus, freezing embeddings reduces performance for the pre-trained model by approximately 12-13 F1 points, while it has no comparable effect on the random model. Under the SEP-only setting, the improvements for Base (+2.7, p=0.004) and Medium (+3.6, p=0.002) are significant, whereas Tiny/Small are not. The frozen-probe condition, however, leaves the token embedding matrix trainable, conflating emergence in transformer weights with task-driven embedding adaptation; we disentangle the two next.

4.3 Representational Changes During Fine-tuning

To understand how representations evolve during fine-tuning, we perform Centered Kernel Alignment (CKA) analysis comparing each fine-tuning epoch against the pre-trained model. CKA analysis shows that full fine-tuning substantially reorganizes representations, with deeper layers diverging most from pre-training (CKA dropping to 0.23 to 0.64; Figure 2a). This raises a key question: is semantic role information *created* during fine-tuning, or already present in pre-trained representations?

4.4 Isolating Emergence from Adaptation

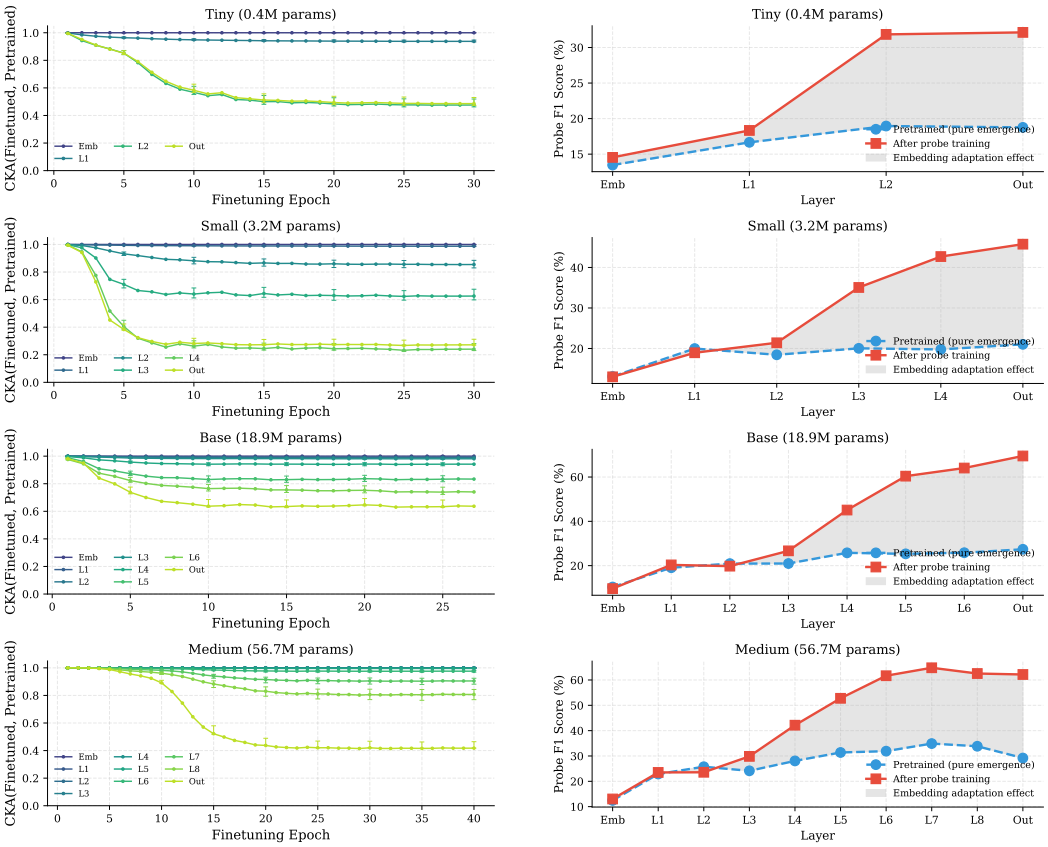
To determine whether semantic role information is encoded during pre-training (emergence) versus created during fine-tuning (adaptation), we train linear probes on pre-trained representations before any task-specific training. We compare this to probes trained after frozen probe training. Semantic role information is present in pure pre-trained representations: linear probes achieve 10 to 35% F1 across layers and scales before any SRL training (Figure 2b). This confirms that emergence occurs even at small scales; predicate-argument structure is encoded during pre-training rather than created during task-specific adaptation. As a stricter control, a frozen-embedding (SEP-only) ablation isolates the residual transformer contribution and is statistically significant at our two largest scales ($p = 0.004$, $p = 0.002$; full results in Appendix I).

4.5 Neuron-Level Dynamics

Overview. We identify role-selective neurons using PCA. First, we run PCA on feed-forward activations across all training rows, then test whether principal components separate semantic roles using ANOVA F-statistics. For each discriminative principal component ($p < 0.0001$), we extract the top 5 neurons with the highest absolute loadings, selecting up to 20 role-selective neurons per layer across all discriminative principal components. We refer to neurons selective for specific PropBank roles: *Agent* neurons (selective for ARG0, the actor/doer of an action), *Time* neurons (selective for ARGM-TMP, temporal modifiers like "yesterday" or "during the meeting"), *Location* neurons (ARGM-LOC), and *Manner* neurons (ARGM-MNR) (PropBank definitions in Section D).

PCA and t-SNE Visualizations. PCA and t-SNE visualizations of final-layer feed-forward activations (Appendix C, Figure 6) reveal that activations show partial organization by PropBank semantic role, with ARG0-Agent and ARGM-TMP showing the clearest separation; the first two principal components explain only 14.8% of variance, indicating that semantic roles are encoded in a higher-dimensional subspace not fully captured by linear projections. *Agent* and *Location* neurons are spread across all four layers of the Small model, whereas *Time* neurons are found mostly in L1–L3 (see Figure 4 in Appendix B), suggesting role encoding is distributed throughout the network. Across all four scales, *Manner* and *Time* neurons are the most selective ($z = 1.1$–2.7) and *Theme* and *Recipient* neurons the least ($z \leq 0.35$) (Table 8). To causally validate neuron importance, we set the weights of PCA-identified neurons to 0 in the *pretrained* model, then fine-tune on QA-SRL and evaluate. This tests whether specific neurons are necessary (not just correlated) for learning semantic roles. We perform role-conditioned neuron ablation: ablating role X's neurons and measuring role X's specific F1 (Table 3).

Role-conditioned Analysis. The role-conditioned results reveal scale-dependent neural organization. Smaller models (Tiny: -0.60 ± 1.75) show clear dependence on *Agent*-selective neurons,



(a) Temporal CKA across layers.

(b) Per-layer probing F1.

Figure 2: Representational changes during fine-tuning, and isolation of pre-training from adaptation. (a) Centered Kernel Alignment (CKA) between each fine-tuning epoch and the original pre-trained model: deeper layers (darker lines) diverge most, dropping to CKA ≈ 0.23 to 0.64, while early layers remain stable. (b) Per-layer linear probes comparing *Pure pre-trained* (blue dashed; before any SRL training) and *After probe training* (red solid). Gray region marks adaptation effect. Probes are trained on 500 samples and measure layer-specific linear separability, so absolute values differ from Table 2. F1 on pure pre-trained representations ranges from 10 to 35% across scales, isolating information present in fixed transformer weights.

while the effect attenuates toward zero at Small (-0.34 ± 2.00), Base (-0.10 ± 2.31), and Medium (-0.25 ± 1.12), with GPT-2 Medium experiencing a sign reversal ($+1.36 \pm 2.41$) (Figure 3). This counterintuitive finding suggests that larger models develop redundant or interfering representations, and that ablating putatively *Agent*-selective neurons removes this interference. *Location* and *Manner* neurons also experience a mild sign-flip across scales (from $+2.83 \pm 3.08$ at Tiny to -0.05 ± 1.86 at Medium for *Manner*, -0.33 ± 2.28 to $+0.74 \pm 2.24$ from Tiny to Small for *Location* neurons), suggesting these role-related neurons are represented distributionally. *Time* neurons experience the opposite sign-flip to *Agent* neurons: ablating them increases *Time* F1 at Tiny ($+0.48 \pm 1.37$, and at every layer in Table 7) but decreases F1 scores at Small (-0.68 ± 0.94) and Medium (-1.13 ± 1.12), suggesting *Time* neurons become causal with scale in our custom models. Combined with the scale-dependent *Agent* effects, this demonstrates that role-selective neurons exist but reorganize with capacity. We find *Agent* neurons become less causal with scale, and find this replicates on GPT-2 (124M, 355M; Appendix E) and Pythia (70M, 160M, 410M; Appendix F) with random-neuron controls confirming role-specificity, and component-level circuit analysis across all six models shows the same concentrated-to-distributed transition (Appendix G; Figure 3). Overall, we find there is a reorganization of which neurons are causally necessary (*Agent* effects attenuate; Figure 3, Appendices E–F), but we do not observe any kind of heterogeneous split sometimes reported in very small

Table 3: Role-conditioned neuron ablation. We ablate each role's selective neurons and measure *that role's* F1 specifically (i.e., F1 computed only on validation rows whose QA-SRL answer has that corresponding role label). Negative Δ confirms causal importance for that semantic role; positive Δ in larger models indicates interference. F1 baselines for each role differ from overall F1 because role-specific subsets are smaller relative to the whole population. The lower block gives overall F1 (all validation rows) and its change when each role's neurons are ablated. In the lower block, F1 is the unablated model's overall F1 and Δ its change after ablation. These baselines are lower than the overall F1 results in Table 2 as our ablation runs fine-tune on 90% of the training rows and keep the epoch with the lowest loss on the held-out 10% of training, whereas Table 2 reports the best validation F1 over all epochs.

	Tiny		Small		Base		Medium	
Role	F1	Δ	F1	Δ	F1	Δ	F1	Δ
Agent	49.3 ± 0.7	-0.60 ± 1.75	52.4 ± 1.3	-0.34 ± 2.00	57.4 ± 1.6	-0.10 ± 2.31	56.5 ± 2.0	-0.25 ± 1.12
Time	54.7 ± 1.4	$+0.48 \pm 1.37$	59.9 ± 1.3	-0.68 ± 0.94	61.9 ± 1.5	-0.12 ± 2.09	65.3 ± 2.1	-1.13 ± 1.12
Location	42.9 ± 2.4	-0.33 ± 2.28	54.5 ± 2.7	$+0.74 \pm 2.24$	57.7 ± 3.2	-0.62 ± 1.09	61.7 ± 2.2	-1.31 ± 0.49
Manner	35.8 ± 3.8	$+2.83 \pm 3.08$	46.2 ± 5.6	-4.27 ± 3.67	60.8 ± 2.2	-6.75 ± 4.52	56.1 ± 2.6	-0.05 ± 1.86
Overall F1 (all validation rows) after ablating each role's neurons								
Agent	41.0 ± 0.6	-0.24 ± 0.55	47.4 ± 1.1	$+0.27 \pm 1.23$	56.1 ± 0.6	-0.58 ± 0.89	56.0 ± 1.3	$+0.06 \pm 0.30$
Time	41.0 ± 0.6	$+0.41 \pm 0.55$	47.4 ± 1.1	$+0.40 \pm 1.10$	56.1 ± 0.6	-0.41 ± 0.50	56.0 ± 1.3	-0.66 ± 0.30
Location	41.0 ± 0.6	-0.30 ± 0.25	47.4 ± 1.1	$+0.85 \pm 1.01$	56.1 ± 0.6	$+0.17 \pm 0.59$	56.0 ± 1.3	$+0.05 \pm 0.38$
Manner	41.0 ± 0.6	-0.02 ± 0.37	47.4 ± 1.1	-0.05 ± 0.39	56.1 ± 0.6	-0.01 ± 0.50	56.0 ± 1.3	$+0.58 \pm 1.08$

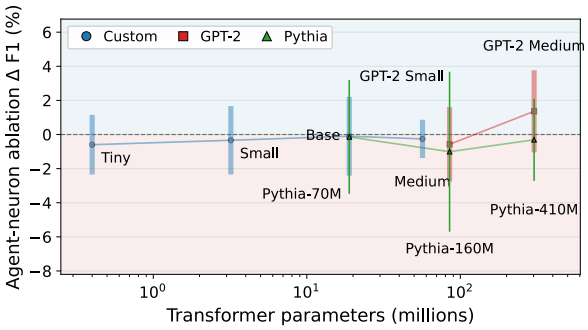


Figure 3: Ablation Δ*Agent* F1 (role-specific, matching Table 3; not overall F1) across 9 parameter configurations. The custom-model trajectory (blue) shifts from causal to ineffective at larger scales; GPT-2 and Pythia follow a similar pattern, with GPT-2's *Agent* neurons becoming interfering at large scales. Error bars: $\pm$ std over 5 seeds. GPT-2 and Pythia values: Table 9.

toy models. We found this surprising relative to previous small-model circuit work that identifies fairly clean specialized components [Olsson et al., 2022, Wang et al., 2023, Conmy et al., 2023].

Cross-role Interference and Co-activation. By ablating role-selective neurons and then fine-tuning, we were able to recover most of the overall F1 (Table 3) across neuron types. Ablating *Agent* neurons changes Medium overall F1 by $+0.06 \pm 0.30$, and ablating *Manner* neurons changes Base overall F1 by -0.01 ± 0.50, against unablated scores of 56.0 ± 1.3 and 56.1 ± 0.6. This recovery shows distributed compensatory mechanisms. In other words, in language models, once *Agent*, *Time*, *Manner*, and *Location* neurons are removed, fine-tuning can recover most of their original function or use built through pre-training. Pairwise correlations between PCA-identified neurons show same-role neurons more correlated than different-role neurons ($\rho_{\text{within}} \approx 0.27$ vs. $\rho_{\text{across}} \approx -0.01$), consistent with distributed, overlapping representations. Per-role critical layers and the power-law fit are in Appendix B; overall ablation F1 is in the lower block of Table 3.

Limitations. *Dataset quality:* the original QA-SRL dataset [He et al., 2015] achieves $\sim$72.4% recall against expert annotations [FitzGerald et al., 2018], which may underestimate absolute F1; our analyses focus on the emergence *gap*, where systematic annotation noise cancels. *Data-constrained scaling:* WikiText-103's $\sim$103M tokens give a tokens-to-parameters ratio of $\sim$1.8 for Medium versus

the Chinchilla-optimal $\sim$20 [Hoffmann et al., 2022]; despite this, Medium achieves the highest fine-tuned F1 ($65.8\% \pm 0.7\%$), and reported emergence is conservative as Medium remains under-converged (Appendix H). *Model scale:* custom models reach 57M transformer parameters; GPT-2 (355M) and Pythia (410M) remain small relative to massive commercialized frontier LLMs. We chose GPT-2 and Pythia because they are fully open source and runnable within our academic budget (under $4,000 total for the project).

5 Conclusion

Summary. We have studied whether semantic role understanding emerges during language model pre-training and presented an investigation of how semantic role understanding emerges in language models across four scales (0.4M to 57M transformer parameters). We have shown that pre-training contributes 12–19 F1 above architecture-matched random initialization across scales, with layer-wise probing on pure pre-trained representations confirming encoding of predicate-argument structure. Moreover, feed-forward neurons activate selectively for semantic roles. Role-conditioned ablations reveal that compensatory mechanisms within language model architectures cause overall F1 to recover after fine-tuning, while *Agent* neurons show scale-dependent effects: causal in smaller models but improving performance when ablated in GPT-2 Medium, suggesting neural reorganization or cross-role interference as model size increases. This scale-dependent reorganization shows that larger models develop complex neuron interactions absent in smaller models.

Implications. Our work has important implications for understanding language models. First, role-selective neurons demonstrate that emergent linguistic capabilities can partially be localized from pre-training, suggesting that interpretability methods can identify meaningful structure in emergent behaviors. Second, cross-role interference in larger models demonstrates that language models do not necessarily implement clean one-to-one mappings between individual neurons and specific functions. Third, the scale-dependent neural reorganization, where *Agent* neurons transition from causally important to interfering, challenges the assumption that circuits identified in smaller models will retain their function at larger scales. Fourth, our frozen probing methodology suggests that semantic role understanding emerges from pre-training on natural language across model scales.

Acknowledgments and Disclosure of Funding

This work investigates the emergence of semantic role understanding through assessing what capabilities are learned during pre-training versus fine-tuning. We see no ethical issues with this work, as understanding how semantic understanding emerges helps demystify LLM capabilities. Our experiments required approximately 860 GPU hours for the custom models plus approximately 200 GPU hours for the GPT-2 and Pythia replications; we have released the full model weights with checkpoints to reduce redundant computation. We thank Modal for a GPU grant that enabled the majority of these experiments to be run at no cost. Carla Griffiths is an Affiliate Researcher at UCL and works full-time for Use Passionfruit Ltd. Use Passionfruit did not fund any of the authors' research and the work was conceived before Carla Griffiths' employment. Mirco Musolesi has no competing interests to declare.

References

Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. In *Computational Linguistics*, volume 28, pages 245–288, 2002.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS'20)*, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 30th Conference on Neural Information Processing Systems (NeurIPS'17)*, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'19)*, 2019.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'18)*, 2018a.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)*, 2018.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'19): Tutorial Abstracts*, 2019.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'19)*, 2019.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'19)*, 2019.

Martha Palmer, Daniel Gildea, and Nianwen Xue. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103, 2010.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *Proceedings of the 12th International Conference on Learning Representations (ICLR'24)*, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Luheng He, Mike Lewis, and Luke Zettlemoyer. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*, 2015.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS'23)*, 2023.

Charlie Snell, Eric Wallace, Dan Klein, and Sergey Levine. Predicting emergent capabilities by fine-tuning. In *Proceedings of the 2024 Conference on Language Modeling (COLM'24)*, 2024.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2019.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*, 2020.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*, 2018b.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP'19)*, 2019.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Daniel Jurafsky. Semantic role labeling using different syntactic views. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 2005.

Jie Zhou and Wei Xu. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL'15)*, 2015.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, 2017.

Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*, 2017.

Michael Roth and Mirella Lapata. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16)*, 2016.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*, 2018.

Peng Shi and Jimmy Lin. Simple BERT models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*, 2019.

Nicholas FitzGerald, Julian Michael, Luheng He, and Luke Zettlemoyer. Large-scale QA-SRL parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)*, 2018.

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 (NAACL'19)*, 2019.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'19)*, 2019.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NeurIPS'13)*, 2013.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, 2014.

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'21)*, 2021.

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 *Proceedings of the 11th International Conference on Learning Representations (ICLR'23)*, 2023.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS'23)*, 2023.

Catherine Olsson, Nelson Elhage, Neel Nanda, et al. In-context learning and induction heads. In *Transformer Circuits Thread*, 2022. Anthropic.

Nelson Elhage, Tristan Hume, Catherine Olsson, et al. Toy models of superposition. *Transformer Circuits Thread*, 2022. Anthropic.

Trenton Bricken, Adly Templeton, Joshua Batson, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. Anthropic.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *Proceedings of the 12th International Conference on Learning Representations (ICLR'24)*, 2024.

Adly Templeton, Tom Conerly, Jonathan Marcus, et al. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024. Anthropic.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR'19)*, 2019.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, et al. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, 2023.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

A Training Hyperparameters

All experiments were conducted on NVIDIA GPUs using Modal's cloud infrastructure. Models were implemented in PyTorch 2.0 with mixed-precision training (FP16) where applicable. We generated the analysis for Figures 3-8 locally using Metal Performance Shaders on an M4 MacBook Pro.

A.1 Compute Resources

Table 4: Compute configuration and estimated training times per model. Base and Medium models were migrated to faster GPUs mid-training due to time constraints (Base: epoch 78, Medium: epoch 68).

Config	GPU	VRAM	Pretrain Time	Finetune Time	Total GPU Hours
Tiny	A10G	24GB	$\sim$48h	$\sim$1h	$\sim$49h
Small	A10G	24GB	$\sim$72h	$\sim$1h	$\sim$73h
Base	A10G $\rightarrow$ H200	24/80GB	$\sim$336h	$\sim$4h	$\sim$340h
Medium	A100 $\rightarrow$ H200	80GB	$\sim$390h	$\sim$10h	$\sim$400h

A.2 Pre-training Hyperparameters

Table 5: Pre-training hyperparameters for each model configuration.

Hyperparameter	Tiny	Small	Base	Medium
Sequence Length	512	512	512	512
Batch Size	64	48	32	96
Gradient Accumulation	1	1	1	1
Learning Rate	5×10^{-4}	2×10^{-4}	2×10^{-4}	1×10^{-4}
Warmup Steps	500	500	1000	2000
Max Epochs	200	200	200	200
Early Stopping Patience	20	20	20	20
Dropout Rate	0.05	0.1	0.1	0.1
Samples per Epoch	200K	400K	800K	1.5M

A.3 Fine-tuning Hyperparameters

All fine-tuning results are reported as mean$\pm$std over 5 random seeds (42, 1042, 2042, 3042, 4042).

Table 6: QA-SRL fine-tuning hyperparameters for each model configuration.

Hyperparameter	Tiny	Small	Base	Medium
Sequence Length	512	512	512	512
Batch Size	64	48	32	32
Learning Rate (Full Finetune)	4×10^{-5}	2×10^{-5}	1×10^{-5}	1×10^{-5}
Learning Rate (Frozen Probe)	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}
Max Epochs	30	30	30	30
Early Stopping Patience	8	8	8	8

A.4 Optimizer Configuration

All models use the AdamW optimizer [Loshchilov and Hutter, 2019] with the following settings:

- $\beta_1 = 0.9$, $\beta_2 = 0.999$.

- Weight decay: 0.01.
- Learning rate schedule: pre-training halves the learning rate when validation loss has not improved for 2 epochs, with early stopping; fine-tuning uses a constant learning rate with early stopping.
- Gradient clipping: Max norm 1.0.
- Label smoothing: $\alpha = 0.1$ (pre-training cross-entropy loss).

A.5 Temporal CKA Analysis Configuration

For the temporal CKA analysis described in Section 3.5:

- Probe set size: 500 examples from QA-SRL validation set.
- Checkpoint frequency: Every fine-tuning epoch.
- Activation pooling: Mean pooling over sequence length (excluding padding).
- CKA variant: Linear CKA with centered activations.

B Neuron-Level Analysis Details

Table 7: Critical layer for each role: layer whose ablation gives the largest *role-specific* ΔF1 (computed on the subset of validation examples carrying that role label, not overall F1). Values in parentheses are role-specific ΔF1 for that layer, in F1 points beyond removing the same number of random neurons from the same layer. Mean over 5 seeds; "none": ablating that role's neurons raises its F1 at every layer; the smallest increase is shown.

	Tiny	Small	Base	Med
Agent	L0 (-7.2)	L0 (-1.7)	L0 (-2.1)	L6 (-1.3)
Location	L0 (-0.5)	L1 (-0.4)	L0 (-1.0)	L0 (-0.7)
Time	none (L0 $+1.9$)	L2 (-2.0)	L5 (-3.0)	L2 (-1.1)
Manner	none (L0 $+0.2$)	L1 (-0.2)	L2 (-0.8)	L1 (-0.5)

Each PCA-identified neuron is assigned the role with the highest z-score (role mean minus overall mean, over the activation standard deviation, on training rows):

Table 8: Top role-selective neuron for each role at each scale, shown as layer/neuron index (z-score). Neurons are ranked by the z-score that assigns each neuron its role (role mean minus overall mean, over the activation standard deviation, on training rows); "–": no neuron has that role.

Role	Tiny (L0–L1)	Small (L0–L3)	Base (L0–L5)	Medium (L0–L7)
Manner	L0/400 (1.52)	L1/739 (2.25)	L1/125 (2.59)	L1/1219 (2.71)
Time	L0/128 (1.14)	L0/665 (1.78)	L5/344 (1.72)	L2/1336 (1.80)
Location	L0/444 (0.12)	L1/210 (1.00)	L1/634 (0.98)	L0/467 (1.00)
Agent	L0/384 (0.68)	L3/398 (0.71)	L5/1998 (0.73)	L6/130 (0.72)
Reason	–	–	L2/1927 (0.99)	L5/1641 (0.47)
Recipient	L1/416 (0.20)	L3/546 (0.35)	L5/406 (0.23)	L3/895 (0.11)
Theme	L1/58 (0.22)	L3/580 (0.14)	L1/1179 (0.11)	L7/726 (0.12)

Table 8 lists the most selective neuron for each role at each scale. This specialization emerges naturally from language model pre-training without explicit role supervision.

B.1 Co-activation Correlation Methodology

The co-activation correlations (ρ) reported in Section 4.5 are computed as follows:

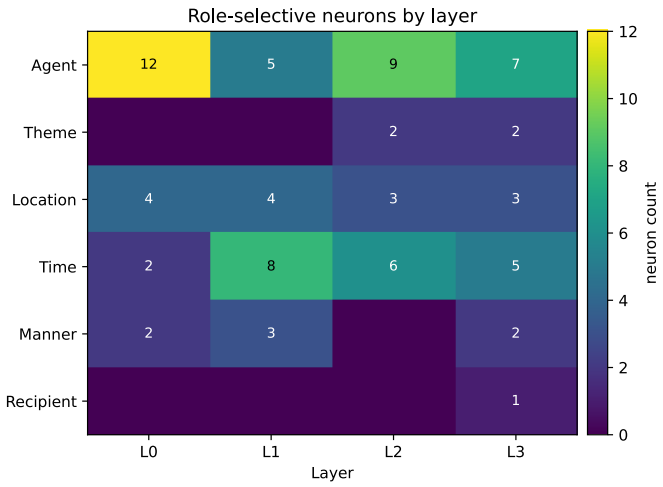


Figure 4: Layer $\times$ role distribution of the top-20 PCA-identified selective neurons per layer in the Small (3.2M) model. Cell values are neuron counts. *Agent* neurons appear in every layer, most at L0 (12); *Time* neurons appear in every layer and peak at L1 (8); *Location* neurons are spread evenly (3–4 per layer); *Manner* neurons are few (0–3 per layer); *Theme* neurons appear only in L2–L3 and *Recipient* only in L3.

1. **Collect activations**: For each QA-SRL validation sample, we extract the feed-forward layer activations $\mathbf{a} \in \mathbb{R}^{d_{\text{ff}}}$ at each layer.
2. **Filter to role-selective neurons**: We retain only neurons identified as role-selective via PCA (see Section 4.5), yielding activation vectors $\mathbf{a}_{\text{selective}} \in \mathbb{R}^{n_{\text{selective}}}$.
3. **Compute pairwise Pearson correlations**: For each pair of role-selective neurons (i, j), we compute the Pearson correlation coefficient across all N validation samples:

$$\rho_{ij} = \frac{\sum_{k=1}^{N}(a_i^{(k)} - \bar{a}_i)(a_j^{(k)} - \bar{a}_j)}{\sqrt{\sum_{k=1}^{N}(a_i^{(k)} - \bar{a}_i)^2}\sqrt{\sum_{k=1}^{N}(a_j^{(k)} - \bar{a}_j)^2}} \tag{6}$$

   where $a_i^{(k)}$ is the activation of neuron i on sample k.
4. **Partition by role relationship**: We separate correlations into two groups based on whether the neuron pair shares the same semantic role:

$$\rho_{\text{within}} = \{\rho_{ij} : \text{role}(i) = \text{role}(j)\} \tag{7}$$
$$\rho_{\text{across}} = \{\rho_{ij} : \text{role}(i) \neq \text{role}(j)\} \tag{8}$$

5. **Aggregate**: We report the mean within-role correlation ($\bar{\rho}_{\text{within}} \approx 0.27$) and mean across-role correlation ($\bar{\rho}_{\text{across}} \approx -0.01$). The difference $\Delta\rho = \bar{\rho}_{\text{within}} - \bar{\rho}_{\text{across}}$ measures functional grouping strength.

C Semantic Role Representations Across Scales

We visualize PropBank-style semantic role representations for all model scales using PCA and t-SNE.

D PropBank Label Assignment

The QA-SRL dataset provides question-answer pairs but not explicit PropBank semantic role labels. To obtain PropBank labels for our analysis, we use AllenNLP's BERT-based SRL model [Shi and Lin, 2019], specifically the `structured-prediction-srl-bert.2020.12.15` checkpoint trained on OntoNotes 5.0. This model achieves an F1 score of 86.5 on the CoNLL-2012 SRL benchmark.

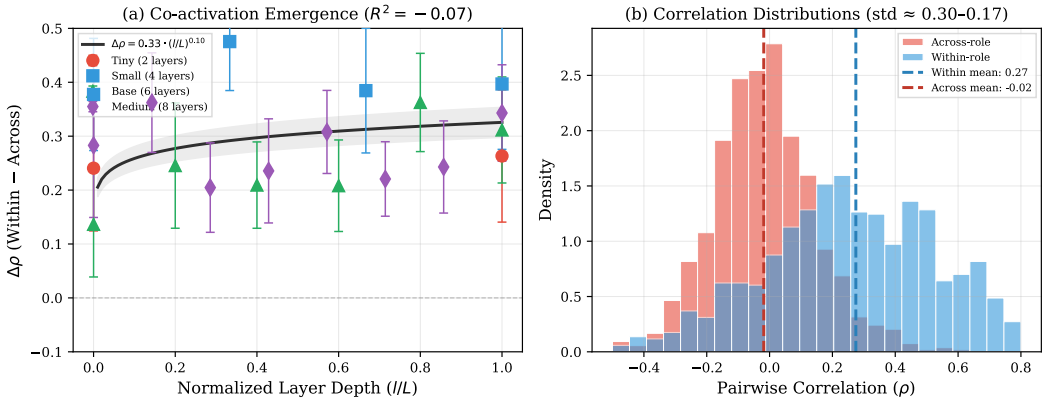
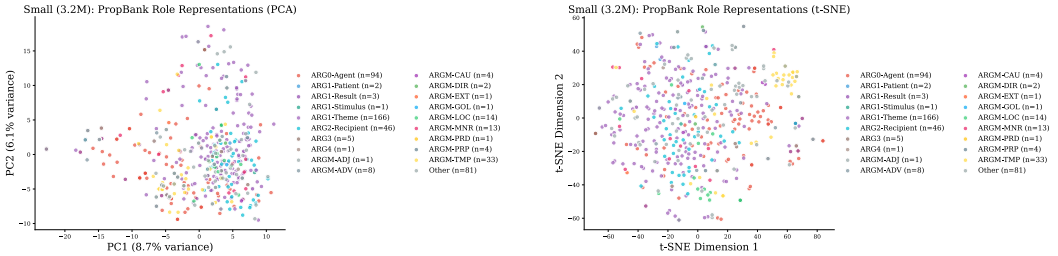


Figure 5: Role-specific neuron separation occurs but does not correlate with increased depth. *Top:* Each point shows $\Delta\rho = \rho_{\text{within}} - \rho_{\text{across}}$ for one layer, measuring whether same-role neurons co-activate more than different-role neurons. Error bars show approximate standard errors. We fit a power-law $\Delta\rho(l) = \alpha \cdot (l/L)^{\beta}$, obtaining $\alpha = 0.33 \pm 0.03$, $\beta = 0.10 \pm 0.05$, $R^2 = -0.07$. *Bottom:* The underlying distributions of within-role (blue) and across-role (red) correlations. The distributions heavily overlap (std $\approx$ 0.15–0.36 vs. mean differences of 0.28), indicating that role-specific groupings are weak but present.



(a) PCA visualization

(b) t-SNE visualization

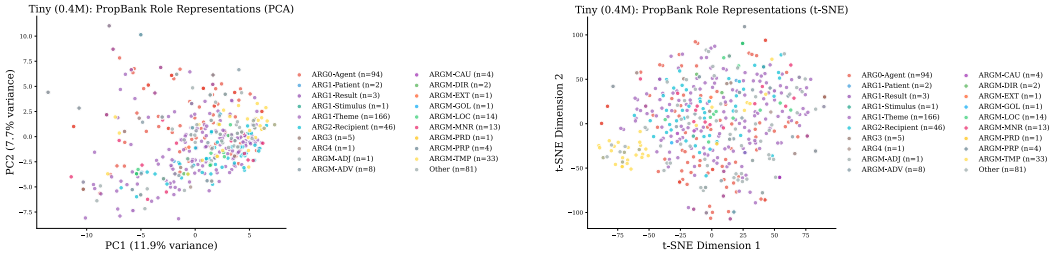
Figure 6: Small model (3.2M transformer parameters): PropBank role representations. Final layer activations colored by PropBank role; ARG0-Agent shows partial clustering, while ARGM-TMP (*Time*) forms a small, distinct cluster for t-SNE dimensions 1 and 2. PCA explains 14.8% of variance in the first two components.

Labeling Process. For each QA-SRL example (sentence, question, answer), we:

1. Run the AllenNLP SRL predictor on the QA-SRL tokenization of each sentence to obtain PropBank argument structure for each verb.
2. Match the QA-SRL answer span to predicted SRL argument spans using QA-SRL's predicate index.
3. Assign the PropBank label if $\geq$50% of the answer overlaps with an SRL argument; otherwise mark as "Other."

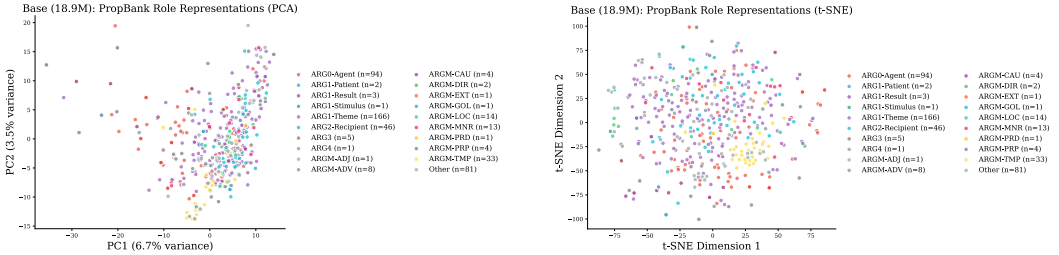
Label Categories. For role-conditioned analyses (correlation, neuron-ablation ΔF1) we collapse PropBank labels into the following readable categories:

- **ARG0-Agent**: Proto-agent (doer, causer).
- **ARG1-Theme**: Proto-patient (undergoer, theme).
- **ARG2-Recipient**: Varies by verb (beneficiary, instrument, attribute, recipient).
- **ARGM-LOC**: Location.
- **ARGM-TMP**: Temporal/Time.



(a) PCA visualization. (b) t-SNE visualization.

Figure 7: Tiny model (0.4M transformer parameters): PropBank role representations. PCA (left) and t-SNE (right) visualization of final layer activations colored by PropBank role. Even at this minimal scale, ARG0-Agent shows some separation. PCA explains 19.6% of variance in first two components, more than larger models, suggesting simpler/more concentrated representations.



(a) PCA visualization. (b) t-SNE visualization.

Figure 8: Base model (18.9M transformer parameters): PropBank role representations. PCA (left) and t-SNE (right) visualization of final layer activations colored by PropBank role. Similar to the Small model, ARG0-Agent shows partial clustering, ARGM-TMP forms a distinct cluster, while other roles (ARG1 variants, most ARGM adjuncts) overlap substantially. PCA explains 10.2% of variance in first two components.

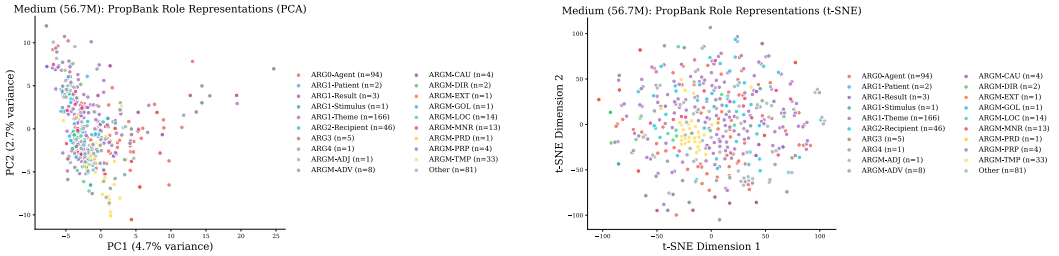
- **ARGM-MNR**: Manner (how the action was performed).
- **ARGM-CAU and ARGM-PRP**: Reason (cause and purpose).
- **Other**: all remaining labels (e.g. ARG3, ARG4, ARGM-ADV, ARGM-DIR, ARGM-PRD).

Raw labels in visualizations. The PCA and t-SNE scatter plots in Appendix C (Figures 6–9) and the role-selectivity heatmap in Figure 4 display the full set of raw labels emitted by AllenNLP rather than the collapsed categories above, in order to expose any sub-role structure to the reader. The additional labels visible in those figures (`ARG2-Recipient`, `ARG3`, `ARGM-ADV`, `ARGM-PRD`, `ARGM-PRP`, and *Theme* and *Recipient* in Figure 4) are AllenNLP's verb-specific or adverbial sub-roles. `ARG2-Recipient` becomes *Recipient*, `ARGM-PRP` becomes *Reason*, and `ARG3`, `ARGM-ADV` and `ARGM-PRD` become *Other* in the collapsed scheme used for all numerical analyses, so the figures and tables refer to slightly different label inventories by design.

E Replication on GPT-2 (Retrain and No-Retrain Ablation)

To rule out that our Agent-neuron results are an artifact of our specific training setup, we replicate the causal neuron ablation protocol on OpenAI's GPT-2 [Radford et al., 2019] at two scales: GPT-2 Small (124M parameters, 12 layers) and GPT-2 Medium (355M parameters, 24 layers). GPT-2 was trained on WebText, providing approximately $80\times$ more tokens per parameter than our custom Medium model and using a different corpus, and is still among the most popular fully open-source decoder-only models.

Protocols used, no-retrain ablation and full re-train. We run two ablation protocols on every model. In *retrain* ablation: weights are zeroed in the pre-trained checkpoint, and then each model



(a) PCA visualization.

(b) t-SNE visualization.

Figure 9: Medium model (56.7M transformer parameters): PropBank role representations. PCA (left) and t-SNE (right) visualization of final layer activations colored by PropBank role. The Medium model shows similar patterns to smaller scales, with ARGM-TMP clustered and ARG0-Agent partially separated, and most other roles distributed across representation space. PCA explains only 7.4% of variance, suggesting representations become more distributed with scale.

is fine-tuned on QA-SRL and evaluated (main figure results; Table 9). In our *no-retrain* ablation: we freeze the pre-trained backbone, train only a QA head (matching the frozen probing protocol), zero the targeted neurons in the resulting probe, and re-evaluate without further training. The two protocols measure related but distinct causal properties: retrain ablation tests whether a neuron is necessary for *learning* the role, while no-retrain ablation tests whether it is necessary for *using* role information already present after probing. We report both signs of the *Agent* effect across families. Random-neuron controls (fourth column of Tables 10 and 11) show near-zero effects under the no-retrain protocol, ruling out trivial perturbation artifacts.

Table 9: GPT-2 and Pythia *retrain* ablation (5 seeds, mean $\pm$ std). Each Δ is the change in that role's F1 after ablating that role's selective neurons, role-specific as in Table 3; the *Agent* column is plotted in Figure 3.

Model	Baseline Agent F1	Agent Δ	Time Δ	Location Δ	Manner Δ
GPT-2 Small (124M)	70.6 ± 1.7	-0.6 ± 2.2	-2.0 ± 0.8	-2.9 ± 1.7	-3.3 ± 4.7
GPT-2 Medium (355M)	72.7 ± 1.4	$+1.4 \pm 2.4$	$+1.0 \pm 1.3$	$+1.6 \pm 1.8$	-2.3 ± 4.0
Pythia-70M	49.3 ± 2.0	-0.1 ± 3.3	-1.4 ± 3.9	-1.7 ± 9.9	$+0.1 \pm 5.7$
Pythia-160M	56.4 ± 4.1	-1.0 ± 4.7	-2.3 ± 5.2	-2.9 ± 4.5	-7.4 ± 4.4
Pythia-410M	75.8 ± 1.3	-0.3 ± 2.4	-0.4 ± 2.5	$+0.6 \pm 3.5$	-2.4 ± 3.0

Table 10: GPT-2 neuron ablation results (5 seeds each). Agent-neurons are strongly causal in GPT-2 Small (-14.1%) and smaller in GPT-2 Medium (-7.5%), matching this attenuation pattern found in our custom models. Random neuron controls show near-zero effects.

Model	Role	No-retrain Δ	Random control Δ
GPT-2 Small (124M)	Agent	$-14.1 \pm 0.7\%$	$+0.1 \pm 1.4\%$
GPT-2 Small (124M)	Time	$-2.9 \pm 1.7\%$	$+0.4 \pm 1.0\%$
GPT-2 Medium (355M)	Agent	$-7.5 \pm 1.6\%$	$+0.4 \pm 0.7\%$
GPT-2 Medium (355M)	Time	$-4.5 \pm 1.2\%$	$+0.3 \pm 1.0\%$

Layer-wise probing on GPT-2 confirms a depth gradient consistent with Section 4.4: the best probing layer is at 83% depth in GPT-2 Small (layer 10, F1 = 37.5%) and 75% depth in GPT-2 Medium (layer 18, F1 = 45.0%).



Figure 10: Layer-wise probing F1 on GPT-2 Small and Medium. SRL information peaks in deeper layers, with the best layer at 83% depth (Small) and 75% depth (Medium).

F Replication on Pythia (No-Retrain Ablation)

We further replicate on EleutherAI's Pythia [Biderman et al., 2023], a third model family trained on The Pile [Gao et al., 2020], a different corpus from both WikiText-103 and WebText. Pythia provides three additional scales (70M, 160M, 410M).

Table 11: Pythia neuron ablation results (5 seeds each). Agent neuron causal effects attenuate from -7.3% at 70M to -4.2% at 410M. Random controls confirm role-specificity.

Model	Role	No-retrain Δ	Random control Δ
Pythia-70M	Agent	$-7.3 \pm 0.8\%$	$-1.5 \pm 1.1\%$
Pythia-160M	Agent	$-3.1 \pm 0.6\%$	$-0.9 \pm 0.7\%$
Pythia-410M	Agent	$-4.2 \pm 1.0\%$	$-0.5 \pm 0.6\%$
Pythia-410M	Time	$+0.3 \pm 0.7\%$	$-0.3 \pm 1.9\%$

The depth gradient also holds: the best probing layer is at 17% depth (Pythia-70M), 33% depth (Pythia-160M), and 42% depth (Pythia-410M).

G Component-Level Circuit Analysis

To characterize the transition from concentrated to distributed organization, we zero-ablated each attention layer and MLP layer individually (5 seeds each) across all six models and measured the causal effect on Agent F1.

H Emergence During Pre-training

We probed intermediate pre-training checkpoints (every 5th epoch) across all four custom scales to track when SRL information appears.

I Frozen-Embedding (SEP-only) Ablation

To isolate what is encoded in pre-trained transformer weights, we freeze the entire token embedding matrix except `[SEP]` (kept trainable so the model can disambiguate question from sentence), alongside positional embeddings and transformer blocks; only `[SEP]` and the QA head update.

Table 12: Component-level circuit analysis. The number of components (attention + MLP layers) needed for 80% of total causal importance increases with scale, while the concentration ratio decreases.

Model	Components for 80%	Total components	Concentration ratio
Custom Tiny (0.4M)	2.8 ± 0.4	4	$70 \pm 10\%$
Custom Small (3.2M)	3.8 ± 0.4	8	$48 \pm 5\%$
Custom Base (18.9M)	6.0 ± 0.6	12	$50 \pm 5\%$
Custom Medium (57M)	8.0 ± 0.6	16	$50 \pm 4\%$
GPT-2 Small (124M)	10.8 ± 0.4	24	$45 \pm 2\%$
GPT-2 Medium (355M)	21.0 ± 1.7	48	$44 \pm 3\%$

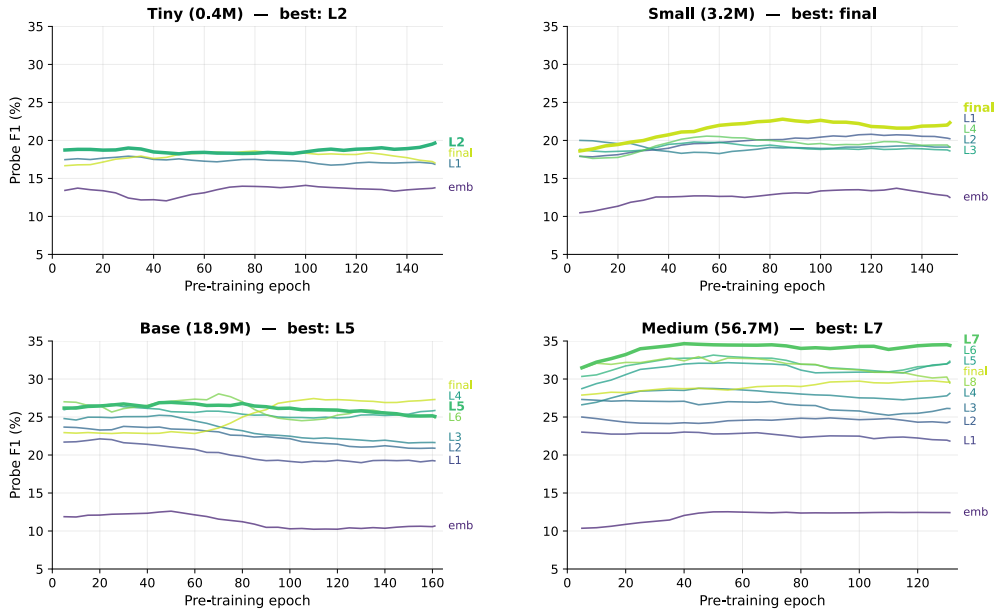


Figure 11: Per-layer probe F1 across pre-training, one panel per scale. Each line is one layer (embeddings, transformer blocks, final norm), labeled at its right endpoint; the best layer per scale is bolded and named in the panel title. F1 stabilizes within the first 10–20 epochs and remains roughly flat thereafter, with embeddings persistently $\sim$10 F1 below transformer-block layers. Across panels, the best layer deepens with scale (L2 in Tiny, final in Small, L5 in Base, L7 in Medium), and steady-state F1 grows monotonically with scale ($\sim$19% / $\sim$21% / $\sim$26% / $\sim$34%).

Freezing the token embeddings decreases the gap between pre-trained and random models from 12–19 F1 (Table 2) to 3–4 F1 for Base and Medium. This gap is still significant at Base ($p = 0.004$) and Medium ($p = 0.002$), indicating that semantic role information is indeed encoded in the pre-trained transformer weights themselves even when all embeddings except the `[SEP]` token are frozen. For Tiny and Small, we find that the gap sizes are insignificant, likely because routing task information through a single trainable token across only 2–4 transformer layers caps what such small models can accomplish; the per-layer probes (Section 4.4), which do not adapt the model at all, still recover above-chance F1 at these scales.

Table 13: Pre-training convergence status. The Medium model has not converged at its final checkpoint, suggesting our emergence scores are conservative lower bounds.

Model	Final best layer	Probe F1	Status
Tiny (0.4M)	block_2	20.4%	Stable from early training
Small (3.2M)	final_norm	23.0%	Slowly rising
Base (18.9M)	final_norm	27.6%	Stable by mid-training
Medium (57M)	block_7	34.8%	Not converged at epoch 131

Table 14: Frozen-embedding (SEP-only) ablation, 5 seeds. With the token embedding matrix held fixed except `[SEP]`, the residual transformer contribution grows monotonically with scale and is significant at Base ($p = 0.004$) and Medium ($p = 0.002$); Tiny/Small are non-significant, reflecting a capacity bottleneck on routing task information through a single trainable token across only 2–4 transformer layers.

Scale	Pretrained F1	Random F1	Emergence
Tiny	23.9 ± 1.0	24.4 ± 1.3	$-0.5 \pm 1.1\%$ (ns, $p = 0.38$)
Small	27.9 ± 1.1	27.4 ± 0.7	$+0.5 \pm 1.9\%$ (ns, $p = 0.59$)
Base	30.5 ± 0.8	27.8 ± 0.4	$+2.7 \pm 1.0\%$ ($t = 6.2$, $p = 0.004$)
Medium	31.6 ± 0.4	28.0 ± 0.7	$+3.6 \pm 1.1\%$ ($t = 7.5$, $p = 0.002$)

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the sign-reversal finding, replication across three model families, and the controlled experimental design, all supported by results in Sections 4–5 and Appendices E–G.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Limitations paragraph (Section 4) discusses dataset quality, data-constrained scaling, model scale limitations, and ablation methodology caveats.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This is an empirical paper with no theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Full hyperparameters in Appendix A, compute resources in Table 4, random seeds specified, and code will be released upon publication.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We use publicly available datasets (WikiText-103, QA-SRL). Code and model checkpoints are available at `https://github.com/carlacodes/emergent-semantic-roles` and `https://huggingface.co/cgriff00/emergent-semantic-roles-checkpoints`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Section 3 describes implementation details and Appendix A provides full hyperparameter tables for pre-training and fine-tuning.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Frozen probe / random / full fine-tuning F1 (Table 2), custom-model role-conditioned and overall neuron ablations (Table 3), GPT-2 and Pythia neuron ablations (Tables 9, 10, 11), the component-level circuit analysis (Table 12) and the SEP-only ablation (Table 14) report mean$\pm$std over 5 random seeds. Critical-layer effects (Table 7) are means over 5 seeds. The SEP-only ablation also reports paired t-tests with p-values.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A, Table 4 details GPU types, VRAM, and total GPU hours per model ($\sim$862 hours for custom models, $\sim$200 hours for replications).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work studies how linguistic representations are organized in language models. We see no ethical concerns.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: This is a scientific study of neural network representations with no direct negative societal applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: Our models are small (up to 57M parameters for custom, up to 410M for replications) and trained on public data. They pose no meaningful risk of misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: WikiText-103 is released under CC-BY-SA 3.0. QA-SRL is publicly available. GPT-2 weights are released under MIT license. Pythia weights are released under Apache 2.0.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will release model checkpoints and analysis code under an open-source license upon publication.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing or human subjects research was conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects research was conducted.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are the subject of study, not a component of the methodology. We do not use LLMs for data generation, annotation, or any non-standard methodological purpose.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.