Fine-tuned network relies on generic representation to solve unseen cognitive task

Dongyan Lin [1 2]

Abstract

Fine-tuning pretrained language models has shown promising results on a wide range of tasks, but when encountering a novel task, do they rely more on generic pretrained representation, or develop brand new task-specific solutions? Here, we fine-tuned GPT-2 on a context-dependent decision-making task, novel to the model but adapted from neuroscience literature. We compared its performance and internal mechanisms to a version of GPT-2 trained from scratch on the same task. Our results show that fine-tuned models depend heavily on pretrained representations, particularly in later layers, while models trained from scratch develop different, more task-specific mechanisms. These findings highlight the advantages and limitations of pretraining for task generalization and underscore the need for further investigation into the mechanisms underpinning task-specific fine-tuning in LLMs.

1. Introduction

Pretraining large language models (LLMs) on vast amounts of data has shown promising results, revealing emerging capabilities such as mathematical skills, translation, and reasoning. This extensive pretraining helps models develop useful representations that allow for subsequent fine-tuning on specific, unseen tasks, thereby broadening the range of problems LLMs can solve. However, it remains unclear whether a pretrained model, when addressing specific tasks, relies on generic pretrained representations that are broadly applicable to most language-based tasks or if it constructs new, niche representations uniquely suited to the target task, akin to optimizing the network from scratch for that task.

In this work, we investigate the ability of pretrained GPT-2 to solve a context-dependent decision-making problem based on numerical comparison through fine-tuning. This task is adapted from neuroscience and cognitive science literature, commonly used to study decision-making in animals and recurrent neural network (RNN) models of the brain, but it is entirely novel to GPT models as it is not present in English text datasets. We aim to understand the extent to which fine-tuned models depend on their pretrained representations to solve a novel task. To this end, we compare the representations after fine-tuning with those developed by GPT-2 optimized solely on this task from scratch. We chose this task not only because it is novel but also because its grounding in neuroscience allows us to explore the data with computational neuroscience methods and make direct comparisons between representations in biological and artificial neural networks.

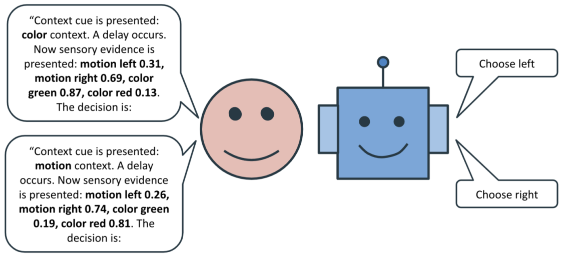
“Context cue is presented: **color** context. A delay occurs. Now sensory evidence is presented: **motion left 0.31, motion right 0.69, color green 0.87, color red 0.13**. The decision is:
“Context cue is presented: **motion** context. A delay occurs. Now sensory evidence is presented: **motion left 0.26, motion right 0.74, color green 0.19, color red 0.81**. The decision is:
Choose left
Choose right

Figure 1. Task schema. Each trial of the context-dependent decision-making (CDDM) task is converted to text description and prompted to GPT-2. The model is trained to respond based on the context cue and sensory evidence, similar to the animal behavioral task.

2. Related Work

Fine-tuning pretrained transformer models enables a variety of capabilities. Fine-tuning pretrained transformer models has demonstrated a wide range of capabilities. For instance, LLMs such as GPT-3 can be fine-tuned to perform specific tasks like text summarization, question answering, and code generation with remarkable proficiency (Brown et al., 2020; Radford et al., 2019). This adaptability is largely attributed to the rich representations developed during the extensive pretraining phase, which captures various syntactic and semantic aspects of language (Devlin et al., 2019; Raffel et al., 2020). Fine-tuning enables these models to leverage their broad knowledge base and adapt to niche tasks with relatively few task-specific examples (Brown et al., 2020; Raffel et al., 2020). Moreover, fine-tuned mod-

[1] Mila [2] McGill University, Montreal, Canada. Correspondence to: Dongyan Lin <dongyan.lin@mail.mcgill.ca>.

ICML Workshop on Large Language Models and Cognition, Vienna, Austria. 2024. Copyright 2024 by the author(s).

els have shown effectiveness in domains beyond natural language processing, such as protein structure prediction and mathematical theorem proving, highlighting their versatility and potential for cross-disciplinary applications (Jumper et al., 2021; Polu & Sutskever, 2020).

Transformers' relation to neuroscience. In addition to their capability and scalability, transformers (Vaswani et al., 2017) have been shown to bear resemblance to traditional computational neuroscience models, namely recurrent neural networks. For example, Whittington et al. (2022) showed that transformers with recurrent position encodings are similar to Hopfield-network-based models of the hippocampus, and exhibited spatial representations such as place cells and grid cells. Zucchet et al. (2023) showed that linearized self-attention can be implemented in linear gated recurrent neural networks. More recent work in neurobiology even suggested that self-attention could in principle be implemented by the brain through cortical waves (Muller et al., 2024) or Hebbian plasticity (Ellwood, 2024). As such, the idea that transformer-based LLMs can be operating on similar principles as human intelligence is not so far-fetched. These parallels also inspired our approach of analyzing transformer hidden states with standard neuroscience data analyses.

3. Experimental Setup

We converted the perceptual random-dot task, which is commonly used to study decision-making in higher cortical areas such as prefrontal cortex (Mante et al., 2013), to text description as the input to GPT-2 (Figure 1A). In each trial, depending on the context cue given at the beginning of the trial, the model is required to choose the left or right target matching the prevalent sensory evidence in either motion (if the context cue is motion) or color (if the context cue is color). As such, it requires attending to the context throughout the trial, and comparing the numerical values for the sensory evidence. Consistent with the neuroscience literature, the magnitude of the motion and color evidence in either direction is randomly controlled by two latent variables, motion coherence coh_m and color coherence coh_c, which vary from trial to trial and are sampled from uniform distribution $U(-B, B)$. For each coherence level, the motion and color inputs are given by:

$$v_{\text{motion, left}} = \frac{1 - \mathrm{coh}_m}{2}, \quad v_{\text{motion, right}} = \frac{1 + \mathrm{coh}_m}{2} \tag{1}$$

$$v_{\text{color, green}} = \frac{1 - \mathrm{coh}_c}{2}, \quad v_{\text{color, red}} = \frac{1 + \mathrm{coh}_c}{2} \tag{2}$$

B ("bound") is used as a hyperparameter; given a fixed number of training samples, the smaller the bound is, the more likely the model will be trained on the same prompt more than once. Under these definitions, positive motion and color coherence provide evidence for the right choice, and negative motion and color coherence provide evidence for the left choice. This means that the model should respond "Choose left" if evidence is stronger in left (if motion context) or green (if color context), and "Choose right" if evidence is stronger in right (if motion context) or red (if color context). We used GPT-2 ("GPT2LMHeadModel") from the Hugging Face library (Radford et al., 2019), which is characterized by its 12 layers, each containing 12 attention heads, and an embedding dimension of 768, consisting of 117M parameters This model is able to produce rich behavior and representations under supervised fine-tuning, but lacks the scale and hence ability to learn this task with in-context learning.

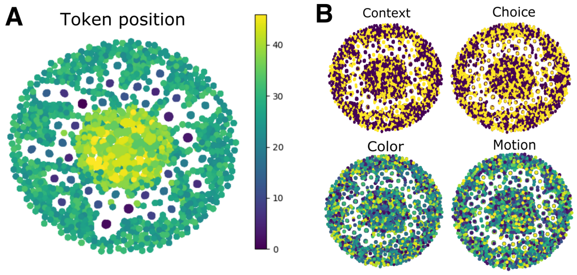

Figure 2. Last layer hidden states from different trials at different tokens after dimensionality reduction by UMAP, colored by token position (A) or different values of each behavioral variable (shown on top of each panel) (B)

We took GPT-2 pretrained on large corpus of English data, and fine-tuned it on this task for a range of training prompts: $\{100, 500, 1000, 2000, 4000, 8000, 10000\}$, and evaluated them on 2000 prompts, during which attention and hidden states data were collected. To test for generalization, we also evaluated fine-tuned models on prompts generated with different bounds (Figure A.1). To understand to what extent fine-tuned GPT-2 relies on pre-trained representations to solve the task, we trained GPT-2 from scratch on the same task for comparison. We found that it took 10 times more training samples to get the same level of performance from a model trained from scratch than from a fine-tuned model. We collected and analyzed attention weights (softmax $\left(\frac{QK^T}{\sqrt{d}}\right)$) and attention outputs (softmax $\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V$) from each head in each layer, as well as the hidden states output from each layer. We applied various analyses to understand the population-level as well as circuit-level mechanisms by which our models solved this task (See Appendix for more details).

4. Results

For the fine-tuning process, we observed that the model achieves higher accuracy (measured by the percentage of correct choice) and lower loss when trained on more samples. Also, smaller bounds on the distribution from which motion and color coherence are sampled leads to better performance, suggesting that seeing repeated examples helped the model learn (Figure A.1 top). We also found that fine-tuned models can generalize well: a model's accuracy when tested on different bounds is closely correlated with its accuracy on training data, meaning that the model learned *how* to perform the task instead of simply memorizing the examples (Figure A.1 bottom).

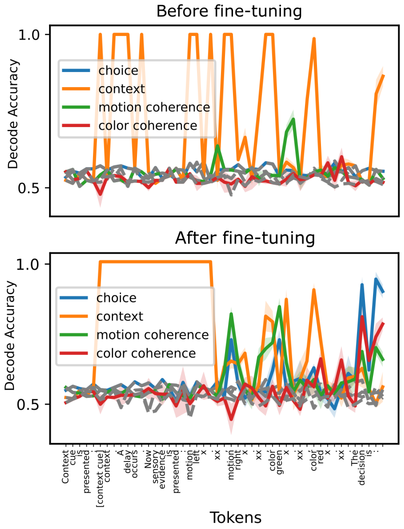

Figure 3. Accuracy of decoding each task-relevant variable from the hidden states at different tokens, for one example unit. Grey dashed lines indicate label-shuffled decoding accuracy as a baseline. Solid lines and shaded area indicate mean and standard deviation of decoding accuracy across 5 cross-validation folds. More example units shown in Figure A.2

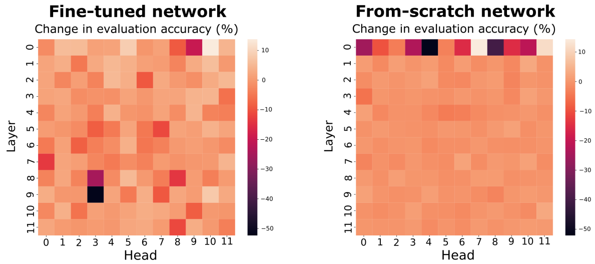

Figure 4. Evaluation accuracy, measured in the percentage of correct repsponses, after zero-ablating each attention head for a pre-trained network fine-tuned on the task (left) and a network trained on the task from scratch (right).

4.1. Population-level structure develops in the hidden states of fine-tuned network.

To understand the key driving factors of representations formed, we projected the hidden states from the last layer at all tokens in all trials to a two-dimensional space with UMAP. We found that the hidden states move in latent space according to the token position: in each prompt, the hidden states start in the middle of the UMAP space, then move outwards, then move inwards. This temporal structure does not apparently correlate to any behavioral variables (context, motion coherence, color coherence, or choice) (Figure 2). We decoded task-relevant variables, including context, motion coherence (whether it's positive or negative), color coherence (whether it's positive or negative), and choice, from the hidden states of the last layer during the presentation of each token with logistic regression before and after tine-tuning (Figure 3; more examples shown in Figure A.2). We notice that, after fine-tuning, a single unit's activation can encode multiple independent task-relevant variables at the same time, similar to the mixed selectivity phenomenon in the brain (Rigotti et al., 2013). Increasing amount of information relevant for decision-making gets encoded as token position progresses.

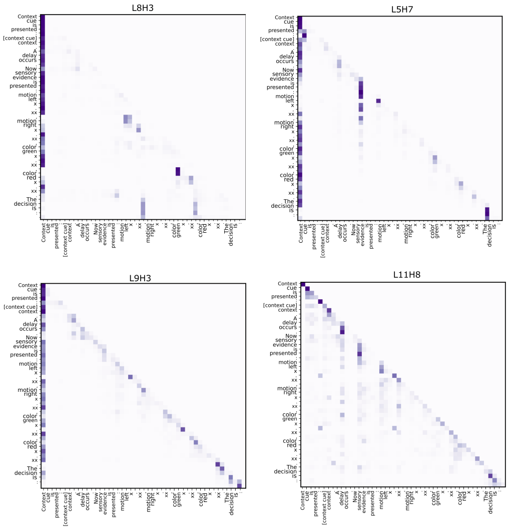

Figure 5. Average attention weights (softmax $\left(\frac{QK^T}{\sqrt{d}}\right)$) across prompts **from a fine-tuned network** for 4 example attention heads that caused significant performance drop when zero-ablated. y-axis is source token, x-axis is destination token. Darker color indicates larger value.

4.2. Task-specific solutions emerge in networks trained from scratch.

To understand whether task-optimized networks and fine-tuned networks converge onto similar circuit-level mechanisms, we ablated each attention head in each layer in trained networks by setting their attention weight to zero, and evaluated the accuracy. We found that fine-tuned networks rely more on attention heads in later layers, particular examples included L5H7, L8H3, L9H3, L11H8, etc, where performance on the task had a significant drop after ablation (Figure 4 left). The importance of these heads is universal across models fine-tuned with different hyperpa-

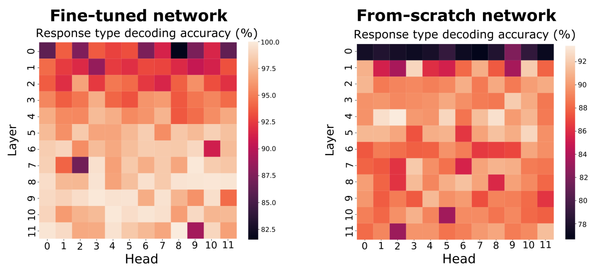

Figure 6. Accuracy of decoding the response type from the output of each attention head (softmax $\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V$), in a pretrained network fine-tuned on the task (left) and a network trained on the task from scratch (right).

rameters (Figure A.3). The trial-averaged attention patterns from these heads did not expose particularly task-specific solutions; instead, we hypothesized that these heads are important for generic language modeling and largely developed their representation during pretraining (Figure 5). Indeed, the same heads in models trained from scratch had not yet developed the same generic attention pattern (Figure A.4). In contrast, networks optimized on the task from random initialization developed more task-specific patterns in their attention weights: drastic performance change happened only when ablating heads from the first layer but not later layers (Figure 4 right). We found that first layer heads pay attention to numbers in the prompt (Figure A.5), which could be a task-specific solution that networks trained from scratch developed.

Next, we asked, what about these heads makes their ablation more detrimental than other heads? To this end, we connected results from head ablation to the representation of behavior in attention outputs. For each head in each layer, we decoded with SVM the type of response produced by the model following the prompt: whether it's choosing left, choosing right, or an invalid response (neither choosing left or right). We found that, for fine-tuned models, heads in later layers indeed showed higher decodability for response type, matching the intuition that they are more important for generic language-based tasks than earlier layers. For models trained on task from scratch, we found that the first layer, whose ablation led to significant performance drop, showed less decodability than other layers. This hints at the possibility that, while heads in the first layer pay significant attention to numbers and other relevant information, the cognitive processing and manipulation of such information to produce appropriate responses could happen in later layers (Figure 6).

5. Discussion

This study demonstrates that fine-tuned models rely more on pretrained representations to solve a novel decision-making task, while models optimized from scratch develop alternative mechanisms. Specifically, the fine-tuned models showed significant reliance on attention heads in later layers, which are likely crucial for generic language modeling, as these heads were developed during pretraining. In contrast, models trained from scratch developed task-specific solutions, with significant performance drops upon ablating heads in the first layer, suggesting that these heads are vital for extracting task-relevant numerical information.

Our findings suggest that while fine-tuned models benefit from the pretrained representations for generalization and performance, models trained from scratch might offer more specialized and potentially efficient solutions for specific tasks. This distinction could be crucial for applications requiring highly specialized problem-solving capabilities. Future research could explore the implications of these findings for the design and deployment of LLMs in various real-world applications, considering the trade-offs between leveraging pretrained knowledge and developing task-specific solutions from scratch.

Our approach of adapting cognitive tasks commonly used in neuroscience and cognitive science laboratories also served another purpose: to evaluate the representation alignment between biological and artificial intelligence, and to advance our understanding of the general principles of intelligence. Complex intelligence systems such as the brain or its computational models often need to pack high-dimensional information from the real world into their activation, making understanding the mechanism by which these activation forms a model of the world a challenge. In the realm of AI, the phenomenon that one neuron can reliably respond to multiple semantic concepts is termed "polysemanticity", and has been observed in vision models and the multi-layer perceptron (MLP) units in language models (Olah et al., 2017; 2020; Elhage et al., 2022; Scherlis et al., 2023). Such phenomenon is similar to the neuroscience phenomenon of "mixed selectivity" in which neurons respond to a combination of sensory inputs, task conditions, and behavioral outputs. Thus, we propose that understanding how LLMs solve cognitive tasks and deciphering the representational alignment between the brain and LLMs would be crucial for the development of more robust and adaptable AI systems.

6. Limitations and next steps

Here we have demonstrated that fine-tuned models relied more on pretrained representations to solve a novel decision-making task, and that models optimized on the same task from scratch which do not have pretrained representations to rely on developed other mechanisms, which we hypothesize to be a combination of first layer extracting numbers and later layers manipulating these information to make a decision. However, further investigation is needed to elucidate the exact mechanism by which models trained from scratch solve this task. Another limitation is that we drew our conclusions based on one cognitive task; further studies with more diverse cognitive tasks are required to understand how pretrained representations support task-specific fine-tuning. Lastly, it is important to develop new quantitative metrics to ensure scientific rigor in our results, as much of our current findings are based on qualitative observations. Similarly, the field of mechanistic interpretability in LLMs, which is also largely qualitative at present, requires new quantitative methods to advance.

Acknowledgements

D.L. is supported by a NSERC Canada Graduate Scholarship—Doctoral (569390-2022). This research was enabled in part by support provided by Calcul Québec (https://www.calculquebec.ca/en/) and Compute Canada (www.computecanada.ca). The author acknowledges the material support of NVIDIA in the form of computational resources.

References

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://papers.nips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019. URL http://arxiv.org/abs/1810.04805. arXiv:1810.04805 [cs].

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. Toy Models of Superposition, September 2022. URL http://arxiv.org/abs/2209.10652. arXiv:2209.10652 [cs].

Ellwood, I. T. Short-term Hebbian learning can implement transformer-like attention. *PLOS Computational Biology*, 20(1):e1011843, January 2024. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1011843. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1011843. Publisher: Public Library of Science.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://www.nature.com/articles/s41586-021-03819-2. Publisher: Nature Publishing Group.

Mante, V., Sussillo, D., Shenoy, K. V., and Newsome, W. T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, November 2013. ISSN 1476-4687. doi: 10.1038/nature12742. URL https://www.nature.com/articles/nature12742. Number: 7474 Publisher: Nature Publishing Group.

Muller, L., Churchland, P. S., and Sejnowski, T. J. Transformers and Cortical Waves: Encoders for Pulling In Context Across Time, January 2024. URL http://arxiv.org/abs/2401.14267. arXiv:2401.14267 [cs].

Olah, C., Mordvintsev, A., and Schubert, L. Feature Visualization. *Distill*, 2(11):e7, November 2017. ISSN 2476-0757. doi: 10.23915/distill.00007. URL https://distill.pub/2017/feature-visualization.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom In: An Introduction to Circuits. *Distill*, 5(3):e00024.001, March 2020. ISSN 2476-0757. doi: 10.23915/distill.00024.001. URL https://distill.pub/2020/circuits/zoom-in.

Polu, S. and Sutskever, I. Generative Language Modeling for Automated Theorem Proving, September 2020. URL http://arxiv.org/abs/2009.03393. arXiv:2009.03393 [cs, stat].

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language Models are Unsupervised Multitask Learners. 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. ISSN 1533-7928. URL http://jmlr.org/papers/v21/20-074.html.

Rigotti, M., Barak, O., Warden, M. R., Wang, X.-J., Daw, N. D., Miller, E. K., and Fusi, S. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585–590, May 2013. ISSN 1476-4687. doi: 10.1038/nature12160. URL https://www.nature.com/articles/nature12160. Number: 7451 Publisher: Nature Publishing Group.

Scherlis, A., Sachan, K., Jermyn, A. S., Benton, J., and Shlegeris, B. Polysemanticity and Capacity in Neural Networks, July 2023. URL http://arxiv.org/abs/2210.01892. arXiv:2210.01892 [cs].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention Is All You Need, June 2017. URL https://arxiv.org/abs/1706.03762v7.

Whittington, J. C. R., Warren, J., and Behrens, T. E. J. Relating transformers to models and neural representations of the hippocampal formation, March 2022. URL http://arxiv.org/abs/2112.04035. arXiv:2112.04035 [cs, q-bio].

Zucchet, N., Kobayashi, S., Akram, Y., von Oswald, J., Larcher, M., Steger, A., and Sacramento, J. Gated recurrent neural networks discover attention, September 2023. URL http://arxiv.org/abs/2309.01775. arXiv:2309.01775 [cs].

A. Appendix

A.1. Training and fine-tuning details

To fine-tune the GPT2LMHeadModel on the CDDM task, we generated a text-based CDDM task dataset, in which each sample follows the template “”Context cue is presented: ... context. A delay occurs. Now sensory evidence is presented: motion left . . . , motion right . . . , color green . . . , color red. . . . The decision is: choose . . . ”, and we concatenated the samples to become the train dataset for the next-token prediction task for the model. We loaded the pretrained checkpoint and tokenizer from huggingface.

To evaluate the model accuracy, we generated new samples either with the same bound for motion and color coherence, or a different bound to test generalization. We asked the model to generate the choice (“Choose left/right”) given the prompt that contains the context cue and the numerical evidence. The evaluation accuracy is defined as the percentage of responses answered correctly.

To train the model on the task from scratch, we used the same training and testing data, as well as a pretrained GPT2 tokenizer, but used newly initialized GPT2LMHeadModel without loading any weights.

For both training and fine-tuning, we ran the experiments with 3 seeds, and grid-searched through the hyperparameter space for training epochs, per-device train batch size, and learning rate. The major results shown in the paper come from the hyperparameter settings in Table 1.

	Fine-tuned model	From-scratch model
Training epochs	12	50
Number of training samples	8000	200000
Per-device train batch size	4	16
Learning rate	5e-5	0.0001
Bound for coherence	0.9	0.7
Seed	2024	2026

Table 1. Hyperparameters used for fine-tuned and from-scratch models

A.2. Analysis

A.2.1. Analysis on last layer hidden states

During the evaluation phase when the model is asked to generate the choice based on the prompt, we collected the hidden states output (dim=768) from the transformer module at each token (in total 47 tokens in the prompt), before they were used to compute the logits.

To project the 768-dimensional vector into a two-dimensional space with UMAP, we stacked the vectors across evaluation prompts across tokens. We color coded the results from dimensionality reduction with token position, context (color or motion), choice (left or right), motion coherence (a number uniformly sampled between -bound and bound), and color coherence (a number uniformly sampled between -bound and bound).

Logistic regression decoding on behavioral variables: we trained a logistic regressor to map the 768-dimensional vector at different token positions in each prompt to the corresponding behavioral variable (context, choice, motion coherence, color coherence) of that trial. We used 5-fold cross-validation, in which in each fold, 80% of the data is used for training and 20% for calculating the mapping accuracy, and averaged across the folds. To obtain a baseline, we shuffled the value of behavioral variables and ran the same analysis.

A.2.2. Analysis on attention weights and attention outputs

Zero ablation: To assess the importance of each attention head to the task performance, during evaluation, we set the attention weights of target head to 0 during forward pass, and evaluate the accuracy on the decision-making task.

During evaluation, at each token position, we collected the attention output from each head in each layer (dim=64) before they were concatenated and used to compute the hidden states output from that attention layer. We trained SVM decoder to decode the response type (choose left, choose right, or invalid response that is either choose left or choose right) from attention outputs across tokens during each prompt (i.e., we concatenated the 64-dimensional vectors from all token positions in a prompt, and used that to decode the type of response to that prompt).

Our analysis was done using scikit-learn (for UMAP, SVM decoding, logistic regression), and we used matplotlib and seaborn libraries to generate our plots. Experiments were done using pytorch and numpy packages. We aim to maximize the reproducibility of this study, and we will release the codebase upon acceptance.

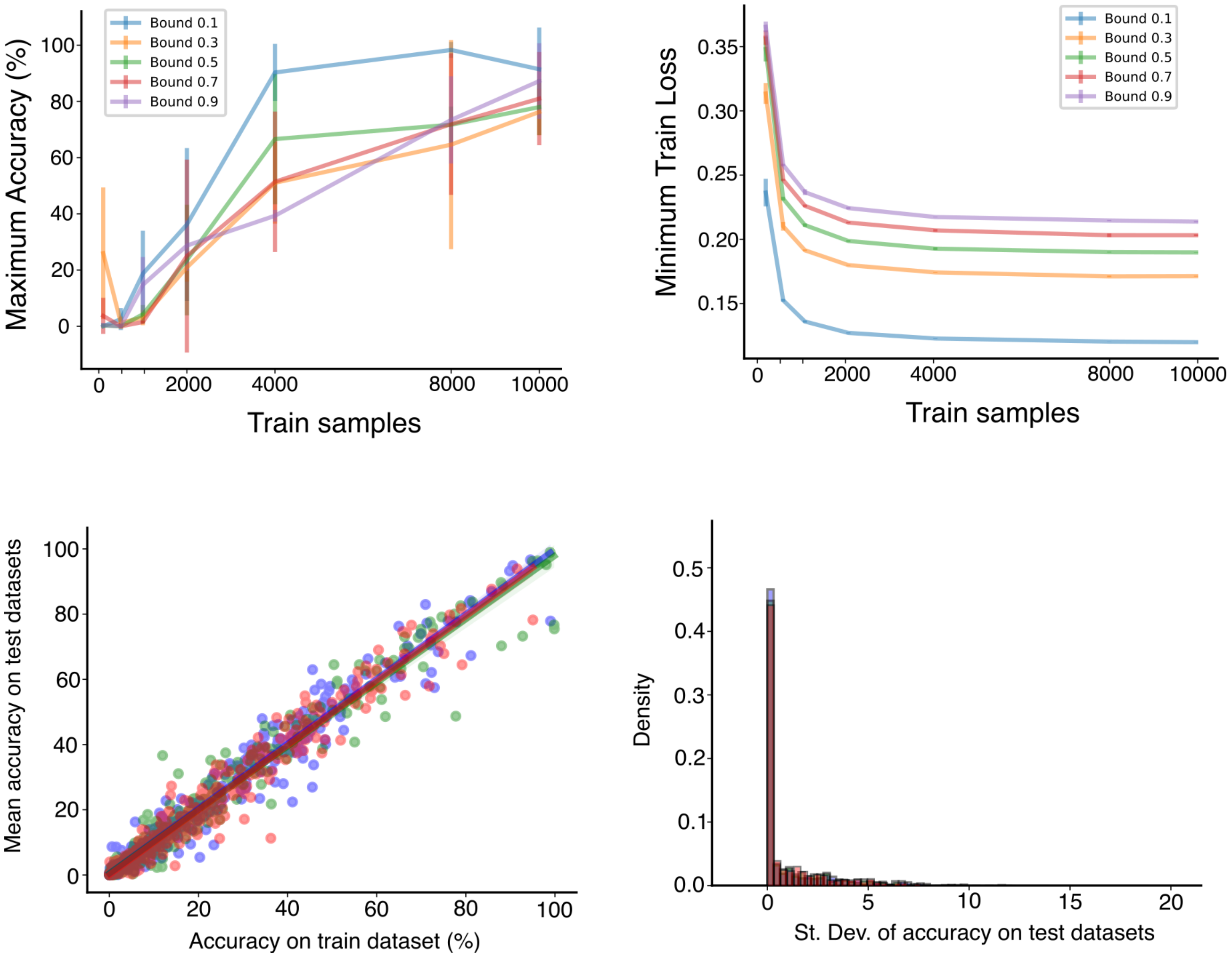

Figure A.1. **Top: Performance of network after fine-tuning.** We plotted the maximum accuracy and minimum loss for each hyper-parameter setting, given the number of trials used for training. Error bar shows mean and standard deviation across 3 seeds. **Bottom: Generalization to test datasets of different bounds. Left:** Mean accuracy on test datasets across the 4 different test bounds that are different from the bound that the model was fine-tuned on, plotted against model's accuracy on the train datasets. Different colors indicate different seeds for fine-tuning. **Right:** Histogram of the standard deviation of accuracy on test datasets across the 4 different test bounds.

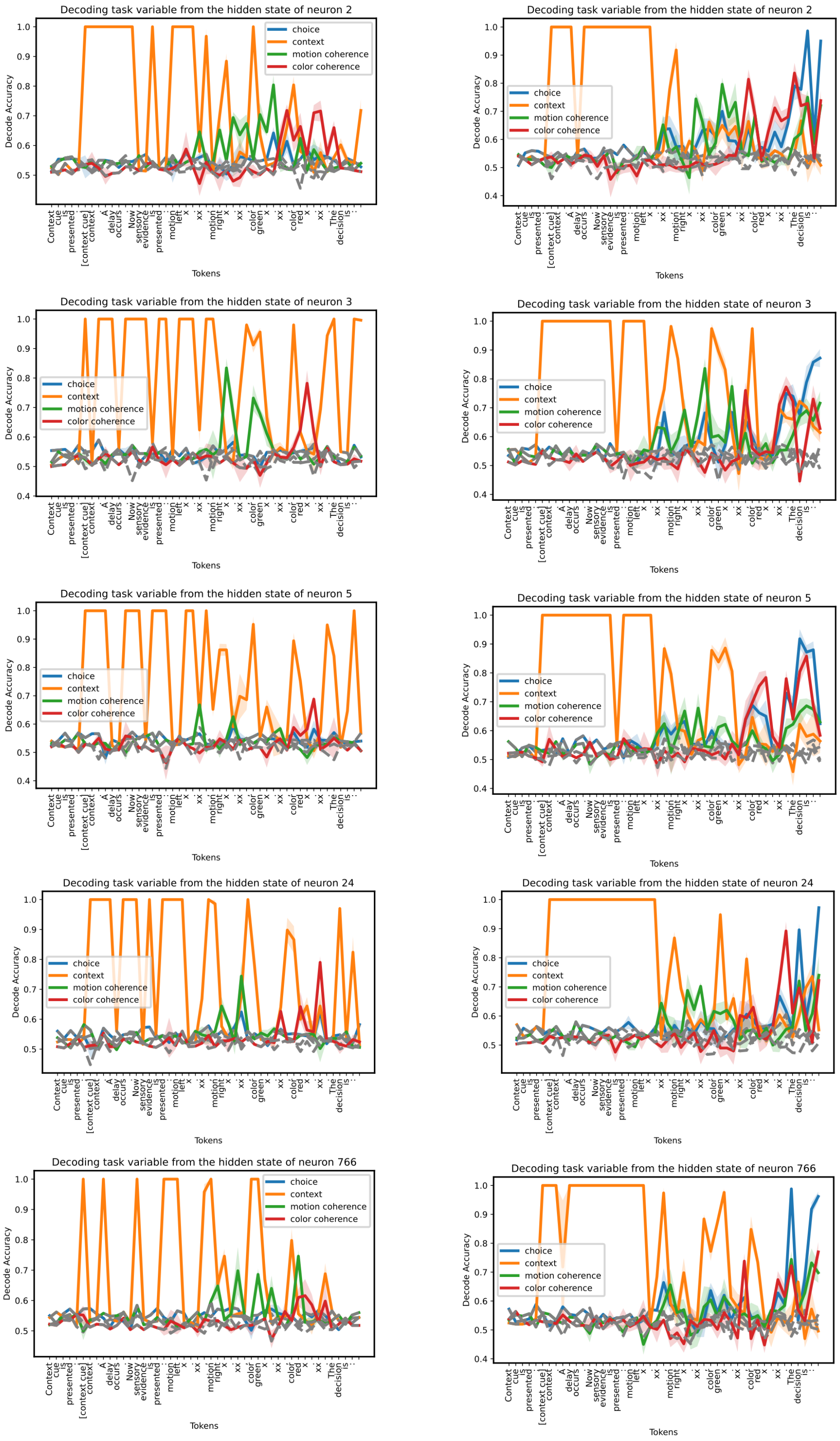

Figure A.2. Behavioral variable decoding (with logistic regression) from last-layer hidden states at each token, for 5 different units (corresponding to the 5 rows). Left column: before fine-tuning. Right column: after fine-tuning.

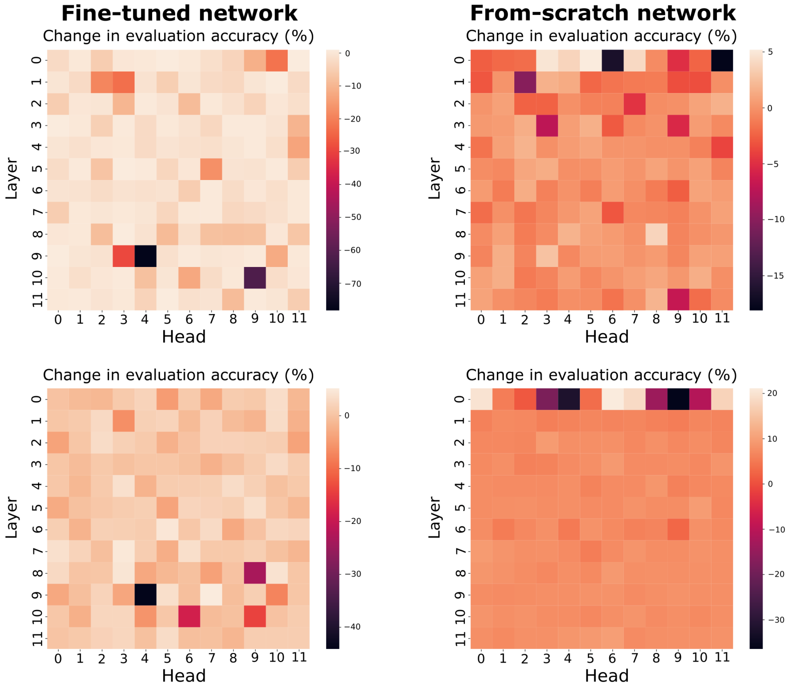

Figure A.3. Supplementary ablation results: Evaluation accuracy, measured in the percentage of correct repsonses, after zero-ablating each attention head, for two more pretrained networks fine-tuned on the task (left) and two more networks trained on the task from scratch (right).

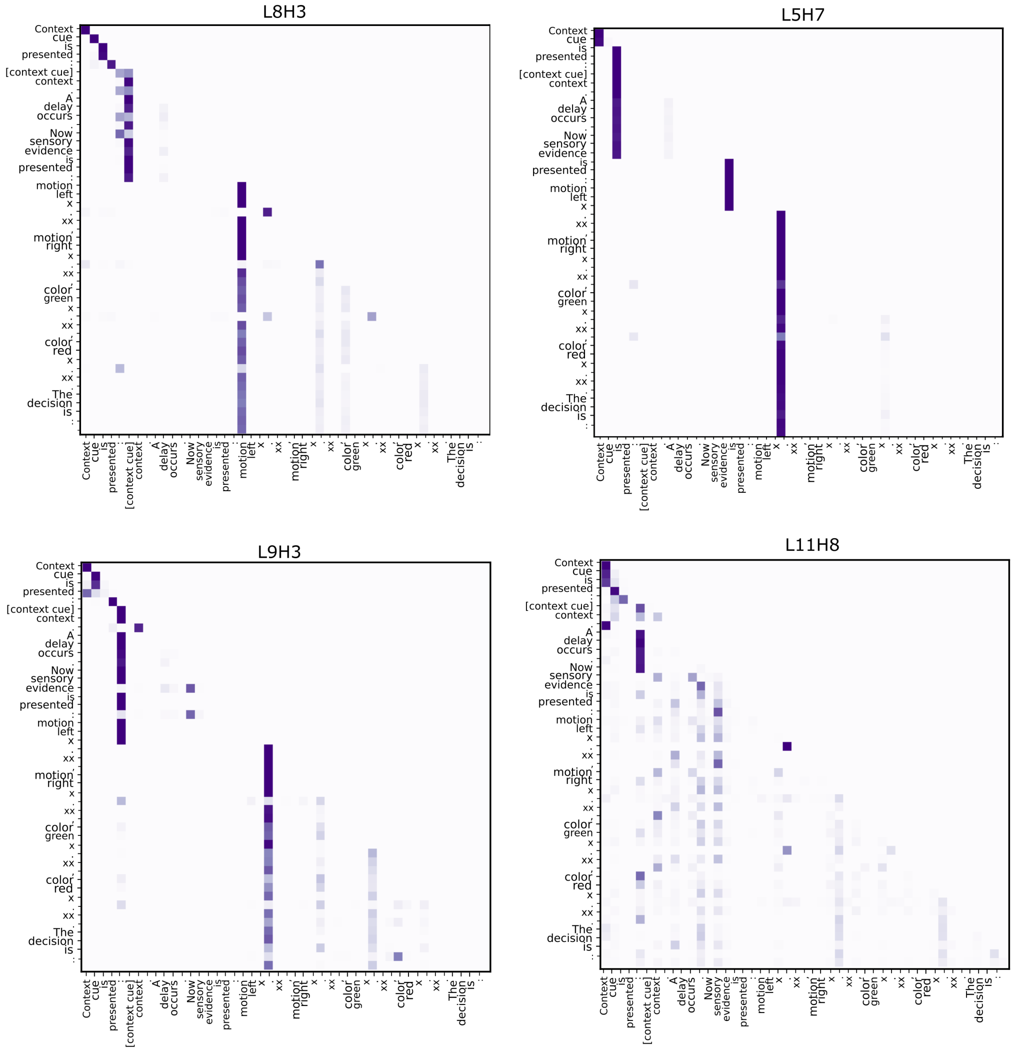

Figure A.4. Average attention weights (softmax $\left(\frac{QK^T}{\sqrt{d}}\right)$) across prompts **from a network trained from scratch** for the 4 example attention heads whose zero-ablation in fine-tuned networks caused significant performance drop. y-axis is source token, x-axis is destination token. Darker color indicates larger value.

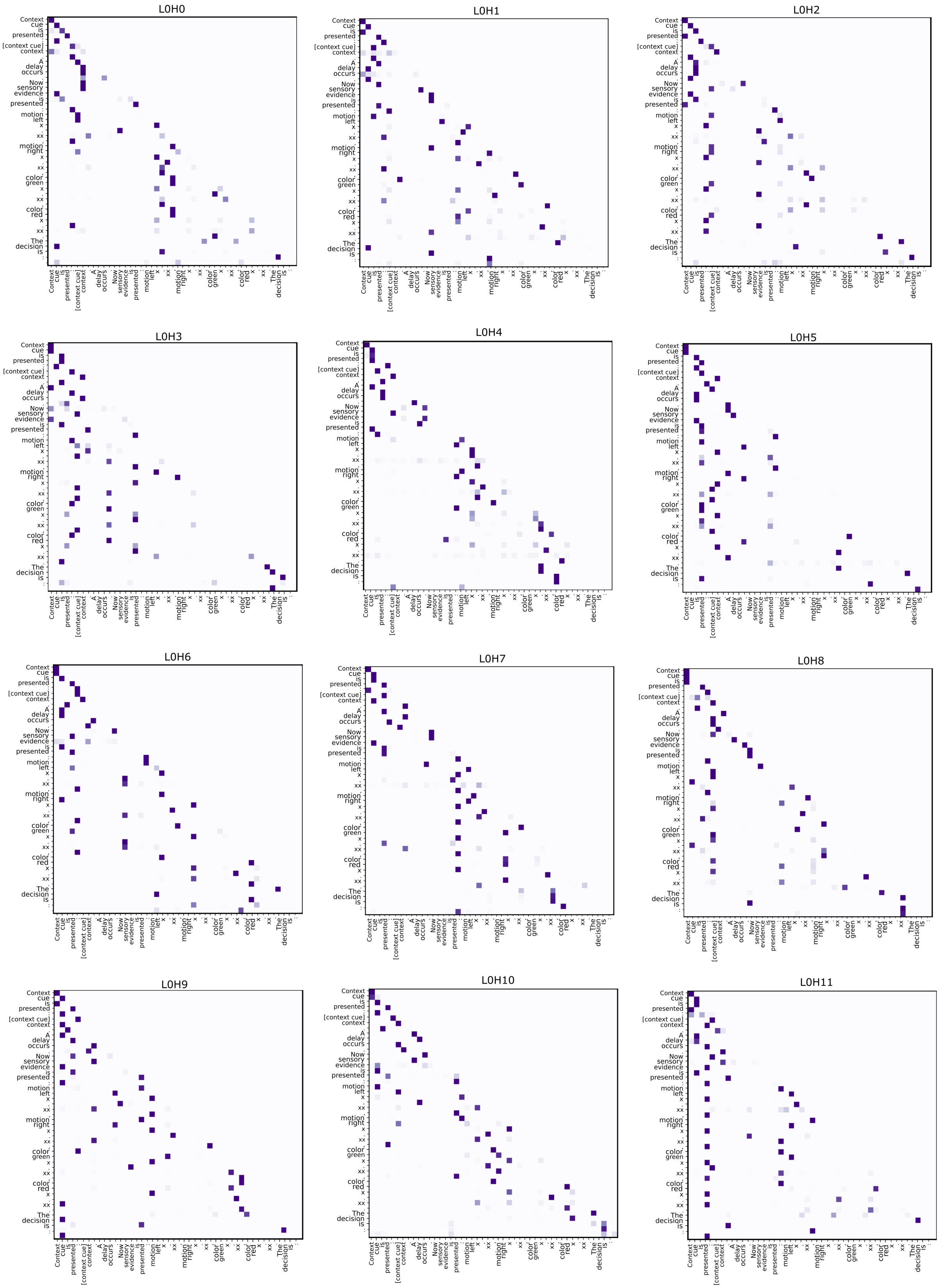

Figure A.5. Average attention weights (softmax $\left(\frac{QK^T}{\sqrt{d}}\right)$) across prompts from a network trained from scratch for the 12 heads in the first layer. y-axis is source token, x-axis is destination token. Darker color indicates larger value.