Structure Development in List Sorting Transformers

Einar Urdshals
ERA
einarurdshals@gmail.com

Jasmina Urdshals
Independent
jasminaurdshals@gmail.com

Abstract

We present an analysis of the evolution of the QK and OV circuits for a list sorting attention only transformer. Using various measures, we identify the developmental stages in the training process. In particular, we find two forms of head specialization later in the training: vocabulary-splitting and copy-suppression. We study their robustness by varying the training hyperparameters and the model architecture.

1 Introduction

Understanding the learning dynamics of neural networks is an important milestone that will aid us in making better predictions for emerging capabilities and enhance our current understanding of a model's inner workings. Using an interesting analogy to biological developmental stages for pluripotent cells, a recent paper by Hoogland et al. [2024] argues for the opportunity to gain insights by adopting a similar mindset for neural networks. Motivated by this approach, we focus on a single layer attention only transformer trained on list sorting. This model has been proposed in McDougall [2023a] and interpreted by McDougall [2023b], and it provides a controlled environment to study the impact of various hyperparameters on the learning dynamics of the model. Using a variety of model-specific and model-agnostic measures, we contribute by:

1. Interpreting the evolution of the QK and OV circuits in transformers during training and identifying distinctive developmental stages.
2. Associating one of these stages, vocabulary-splitting, with a decrease in model complexity.
3. Identifying a new minimal example of copy-suppression.

2 Methods

2.1 Baseline Model Setup and Training

The model is trained with a setup similar to McDougall [2023a] on input sequences of the form [8, 3, 5 SEP, 3, 5, 8], where numbers are sampled uniformly from 0 to 50 and do not repeat, producing a vocabulary size of 52. The model sorts by outputting the next number starting at the separation token, and outputs a list of numbers of the form [x, x, x, 3, 5, 8, x], where the positions marked with x are not included in the loss function. Our baseline model uses list lengths of 10.

We use a single layer attention only transformer model trained to sort lists of numbers. The baseline model architecture includes a residual stream size of 96, 2 attention heads with head dimension of 48, and Layer Normalization (LN) [Ba et al., 2016]. The model is trained with Weight Decay (WD) set to 0.005 using the Adam optimizer [Loshchilov and Hutter, 2019] with a learning rate of 10^{-3}, a dataset size of 150000 with a batch size of 512 and a cross-entropy loss function. The architecture is implemented using `TransformerLens v.2.1.0` [Nanda and Bloom, 2022]. The machinery used for training the models is NVIDIA RTX-4090.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2.2 Measures

We use a variety of measures to study the model. First, we employ the **Local Learning Coefficient (LLC)** introduced by Lau et al. [2023], based on prior work from Singular Leaning Theory (SLT) [Watanabe, 2009]. This measure is discussed in App. A.
In addition, we also use model-specific measures, building on the solution by McDougall [2023b] and targeting the full OV and QK circuits[1]: We define **the Circuit Rank** as the sum of the matrix ranks of the full OV and QK circuits [Elhage et al., 2021]. For an untrained model, the matrix rank is equal to the head dimension (48) for each of the circuits. We also introduce the **Translational Symmetry** and **Head Overlap** to measure the regularity of the model along lines parallel to the diagonal and the overlap of circuits of different heads. For details, see App. B.

3 Background

McDougall [2023b] interpreted a similar[2] model to our snapshot at step 3410, shown in the lower left panel of fig. 4 in the Appendix. They found that the **QK circuit directs the attention of the model: source tokens attend most to the smallest token vocabulary larger than themselves, which results in the higher value band above the diagonal. The OV circuit acts as a copying circuit**, copying forth tokens that are present in the context, as can be seen from the higher values on the diagonal. Together, these circuits bring attention to the smallest token in the context, larger than the current token. Since the context consists of the unsorted list and the sorted list up to and including the current token, the attended to token will be the smallest token in the unsorted list larger than the current token. Additionally, McDougall [2023b] points out the specialization of the attention heads to handle different regions of the vocabulary space. This can be seen from the diagonals of the OV circuits of the different heads, splitting into different regions of vocabulary.

4 Results

We want to investigate how the model learns during training by looking at the evolution of the OV and QK circuits alongside the accuracy, the LLC and the rest of the model specific measures. In Fig. 1 we present the evolution of the baseline 2-head model during training, featuring heatmaps of the circuits for the two attention heads, as well as an upper panel denoting the values of various measures.

Early in the training, at training step 133 (left of Fig. 1) the model learns to sort with **100% accuracy** and the QK and OV circuits develop the expected patterns for the solutions, as discussed in section 3, but there is no clear head specialization yet.

Between training steps 391-3410 the **LLC and the Circuit Rank decrease** simultaneously from their maximal values during training, as the heads specialize by splitting the vocabulary between them (referred to as **vocabulary-splitting** henceforth). For the rest of the training, the model undergoes subtler changes, which we describe in more detail in App. C.2. We show the model at the end of training (step 764225, right of fig. 1). Both circuits feature off-diagonal patterns indicative of the vocabulary-splitting regions. The OV circuits of each head have equidistant rectangular regions. The QK follows suit by keeping decreasing horizontal patterns within the regions where the OV copies. These patterns develop steadily during training, after the decrease of the LLC and Circuit Rank. The translational symmetry increases throughout training, plateauing among other places where the LLC drops, whereas the head overlap peaks as the model learns to sort, after which it drops.

In App. C, we **vary the number of heads and remove LN, WD or both**. When increasing the number of heads, we find that two of the heads still specialize into vocabulary-splitting heads as expected, whereas additional heads seem to specialize into a different mode, which we identify with **copy-suppression** as previously discovered in other models by McDougall et al. [2023] (see Fig. 2).

[1] The full OV and QK circuits are defined as $W_{OV}^h = W_E W_V^h W_O^h W_U$, $W_{QK}^h = W_E W_Q^h \left(W_K^h\right)^T W_E^T$, where W_E and W_U are the embedding and unembedding matrices. W_Q^h, W_K^h, W_V^h and W_O^h are the query, key, value and output matrices of head h, respectively. When referring to the OV and QK circuits in this paper, we mean the full OV and QK circuits. All matrices making up the circuits are learned during training.

[2] The model interpreted by McDougall [2023b] has the same architecture and is trained on the same data (up to a different random seed), but is trained with a different learning rate, and for a different number of steps. The resulting circuits look qualitatively similar.

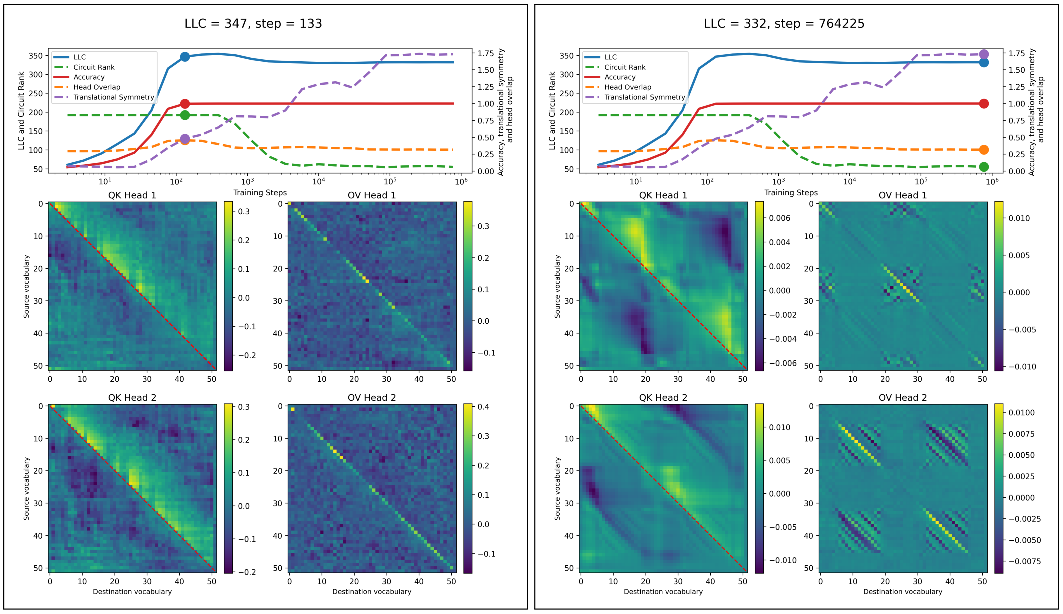

Figure 1: **Baseline 2-head model** as the model reaches 100% accuracy (left) and at the end of training (right). The dashed red diagonal lines in the QK circuit indicates the location of the diagonal

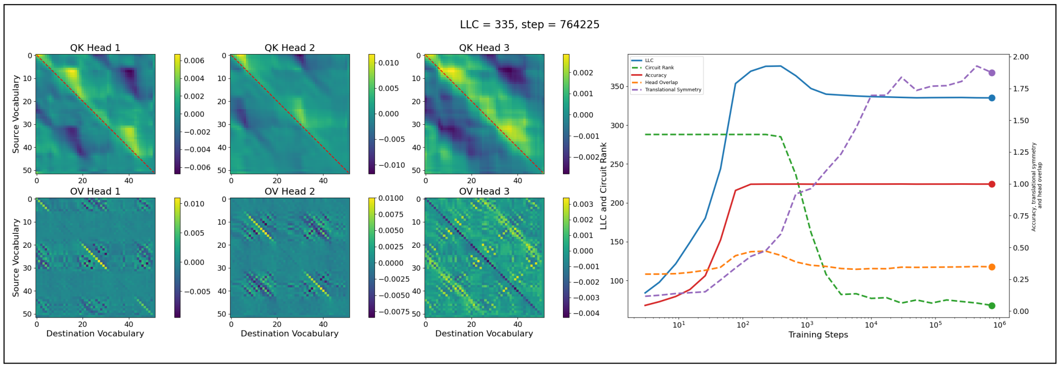

Figure 2: **3-head model** at the end of training. Head 3 performs **copy-suppression** in the OV circuit.

Instead, if we only have 1 head, no specialization is present. We find that removing WD leads to noisier circuits and weaker vocabulary splitting, whereas removing LN causes the model to learn slower.

In App. D we **vary the training data**, such as the size of the vocabulary and the length of the lists, and we experiment with perturbing the training data. Increasing the vocabulary size increases the size of the regions but keeps their number the same as in the baseline model. Increasing the length of the list, on the other hand, increases the number of vocabulary regions. Finally, perturbing the training data causes the model heads to specialize such that their OV circuits perform copying and **copy-suppression** on the entire vocabulary length. Its QK circuit is also different from the expected solution. This is the only setup for which we don't observe a drop in the LLC.

5 Discussion

In the 2-head baseline setup, we observe **three developmental stages**: 1) A learning stage, characterized by increasing accuracy and LLC, 2) an intermediate stage, where both heads attending to and copying overlapping vocabularies and a 3) **head specialization** stage, either into vocabulary-splitting, copy-suppression or both. The intermediate stage is not present if LN is removed during training, but otherwise this trajectory is robust to other variations.

Vocabulary-splitting head specialization is a recurrent feature[3] for this model, even when removing LN, WD, both LN and WD or increasing the number of heads (for details, see Apps. C.3-C.7). It is a **simpler model**, when compared to the preceding stage, where both heads attend to and copy overlapping vocabulary ranges and it is always accompanied by a drop in the LLC. Importantly, the LLC decrease[4] is indicative of a solution that is both simpler *and* performs well on the task at hand, which distinguishes it from other model complexity proxies such as The Circuit Rank. This is exemplified in the 2-head model without LN (see Fig. 7 in the Appendix), where the LLC increases early in the training, as the model significantly simplifies while sorting with only 20% accuracy. It reverses this trend as the transition to head specialization occurs.

The specialization of heads into **copy-suppression** states is qualitatively different between the model trained with perturbed data and the models with more than 2 heads. Their QK circuits look significantly different, so it might be that they are implementing a functionally distinct solutions. For the 3-head and 4-head models, the heads that settle into full vocabulary copying or copy-suppression are preceded by a stage where they are specialized according to vocabulary splitting, in tandem with the other heads. This specialization is not clearly associated with an LLC decrease in any of the setups.

6 Related Work

Hoogland et al. [2024] found developmental stages, including a drop in the LLC corresponding to model simplification, when training a transformer on linear regression. The LLC evolution of non-transformer toy models has previously been studied by Panickssery and Vaintrob [2023] and Chen et al. [2023]. Without using the LLC, Chen et al. [2024] studied developmental stages in BERT. Bagiński and Kolly [2023] and McDougall [2023b] studied algorithmic transformers trained on list sorting, and Nanda et al. [2023] reverse-engineered an MLP trained on modular addition. In this paper, we find copy-suppression, previously observed by McDougall et al. [2023].

7 Limitations

The LLC is only defined at a local minimum, which models during training never are in practice. Lau et al. [2023] argues that the LLC value is not trustworthy, but that the relative ordering of LLCs at different stages of training is. The LLC hyperparameter selection is not rigorous, and we went with the heuristics of seeking parameter space, in which the LLC is locally hyperparameter independent.

Our study is done on a toy model, and one should be careful to generalize our findings to larger transformers. Sporadic experimentation has shown that our results are seed independent, but we have not explicitly checked this. Finally, our interpretation of the functionality of the circuits is approximate, and we expect there is probably more going on in the model.

8 Conclusion

We present a new approach to analyzing the evolution of a model during training, by studying the development of the QK and OV circuits in a list sorting transformer, in tandem with various relevant measures. The developmental stages vary somewhat on the training setup, but a recurring stage is head specialization into vocabulary-splitting, copy-suppression or both.

In particular, vocabulary-splitting is an interesting stage, since it is a simpler model than earlier training stages. It is robust with respect to various changes to the main setup (except for training with perturbed data) and it is well captured by the LLC, which measures model complexity.

The specialization into copy-suppression is observed when perturbing the training data or simultaneously with vocabulary-splitting when training with more than 2 attention heads. They constitute new and minimal examples of this phenomenon, which was first discussed in McDougall et al. [2023]. Further studies could focus on using a similar approach to study the developmental stages of more complicated neural networks that have been interpreted by others. Additionally, one could further investigate the role of the head specializations and the reasons driving their appearance.

[3] When removing WD, vocabulary-splitting is more prominent for vocabulary tokens less than 20.

[4] We also observe an LLC decrease for the 1-head model, see App. C.1, where no vocabulary splitting is possible. We speculate by attributing the simplification to the emergence of off-diagonal features in this model.

9 Statement of Contributions

Einar ran all of the experiments, analyzed and interpreted the results, and produced the figures. Jasmina contributed to analyzing and interpreting the results. The authors wrote the paper together.

Acknowledgments and Disclosure of Funding

Einar's research was supported by the ERA Fellowship. Jasmina's research was supported by an LTFF grant. The authors thank Jesse Hoogland and Rudolf Laine for mentorship and valuable input, and Dan Murfet, Jesse Hoogland, Stefan Heimersheim, and Nicky Pochinkov for valuable feedback on the work.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Mateusz Bagiński and Gabin Kolly. One attention head is all you need for sorting fixed-length lists. https://apartresearch.com, January 2023. Research submission to the research sprint hosted by Apart.

Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MO5PiKHELW.

Zhongtian Chen, Edmund Lau, Jake Mendel, Susan Wei, and Daniel Murfet. Dynamical versus bayesian phase transitions in a toy model of superposition, 2023. URL https://arxiv.org/abs/2310.06301.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Jesse Hoogland, George Wang, Matthew Farrugia-Roberts, Liam Carroll, Susan Wei, and Daniel Murfet. The developmental landscape of in-context learning, 2024. URL https://arxiv.org/abs/2402.02364.

Edmund Lau, Daniel Murfet, and Susan Wei. Quantifying degeneracy in singular models via the learning coefficient, 2023. URL https://arxiv.org/abs/2308.12108.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Callum McDougall. Mech interp challenge: October - deciphering the sorted list model. https://lesswrong.com, October 2023a. URL https://www.lesswrong.com/s/EYjH8M5KLmjuNtJEj/p/frLTfKr8NFv7WCcWG.

Callum McDougall. Mech interp challenge: November - deciphering the cumulative sum model. https://lesswrong.com, November 2023b. URL https://www.lesswrong.com/s/EYjH8M5KLmjuNtJEj/p/uPa63suC8idWhYGbg.

Callum McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. Copy suppression: Comprehensively understanding an attention head, 2023. URL https://arxiv.org/abs/2310.04625.

Neel Nanda and Joseph Bloom. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens, 2022.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability, 2023. URL `https://arxiv.org/abs/2301.05217`.

Nina Panickssery and Dmitry Vaintrob. Investigating the learning coefficient of modular addition: hackathon project. https://lesswrong.com, October 2023. URL `https://www.lesswrong.com/posts/4v3hMuKfsGatLXPgt/investigating-the-learning-coefficient-of-modular-addition`.

Stan van Wingerden, Jesse Hoogland, and George Wang. Devinterp. `https://github.com/timaeus-research/devinterp`, 2024.

Sumio Watanabe. *Algebraic Geometry and Statistical Learning Theory*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2009.

A Singular Learning Theory and the Local Learning Coefficient

Our main tool for studying model development is the Local Learning Coefficient (LLC), a theoretically well-motivated measure of model complexity defined by Lau et al. [2023]. It is based on the learning coefficient from Singular Learning Theory (SLT) [Watanabe, 2009].

The LLC is a measure of the degeneracy of the loss landscape near a model's parameters w^*, where a lower LLC indicates a more degenerate and less complex model. Given an empirical loss $\ell_n(w)$ over parameters w, we calculate the LLC estimate at a local minimum w^* similar to Hoogland et al. [2024] and Lau et al. [2023]:

$$n\beta \left[\mathbb{E}^{\beta}_{w|w^*,\gamma}[\ell_n(w)] - \ell_n(w^*)\right],$$

where $\mathbb{E}^{\beta}_{w|w^*,\gamma}$ denotes the expectation with respect to a tempered posterior distribution centered at w^*, β is an inverse temperature, and γ controls the localization around w^*. Sampling this posterior is done via Stochastic Gradient Langevin Dynamics (SGLD).

The LLC is calculated using the `DevInterp v.0.2.2` software package [van Wingerden et al., 2024]. The hyper-parameters vary with the setup, and are found by performing parameter scans, where we look for regions of parameter space where the LLC is hyper-parameter independent. The LLC of the baseline 2-head model has been calculated with inverse temperature $n\beta = 512/\ln 512 \approx 82$, step size $\epsilon = 3 \times 10^{-5}$, localization term $\gamma = 56$, $n_{\text{chains}} = 4$ and $n_{\text{draws}} = n_{\text{burnin}} = 30000$. The machinery used to calculate the LLC is NVIDIA RTX-4090.

B Translational Symmetry and Head Overlap

Translational Symmetry is introduced to measure the irregularity of the OV and QK circuits in lines perpendicular to the diagonal vs parallel to the diagonal. For a given head h and circuit c, this measure is given as

$$\mathcal{S}_T^{c,h} = \frac{\sum_{ij} |W^{c,h}_{i,\,j} - \frac{W^{c,h}_{i-1,\,j+1} + W^{c,h}_{i+1,\,j-1}}{2}| - |W^{c,h}_{i,\,j} - \frac{W^{c,h}_{i-1,\,j-1} + W^{c,h}_{i+1,\,j+1}}{2}|}{\overline{|W^{c,h}|}\sum_{ij}},$$

where $\overline{|W|}$ denotes the average absolute value of the whole matrix and the sum is taken over all elements not on the edge of the matrix. $\sum_{ij}$ in the denominator denotes the number of such elements. The translational symmetry shown in the plots is the symmetry summed over all circuits and heads (where the number of heads is denoted by N_h):

$$\mathcal{S}_T = \sum_{c \in \text{OV,QK}} \sum_{h=1}^{N_h} \mathcal{S}_T^{c,h}$$

If the circuit is perfectly translationally symmetric, the irregularity of lines parallel to the diagonal will be 0. List sorting is translationally symmetric away from the vocabulary boundary, as a sorting

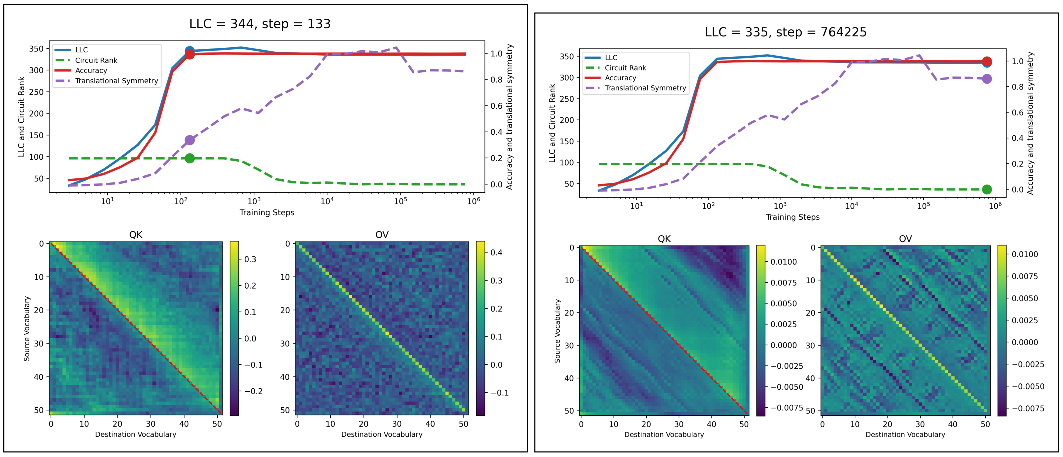

Figure 3: **1-head** list sorting model. As the LLC and Circuit Rank drop, off-diagonal patterns appear in the OV circuit (right) that are not present as the model learns how to sort well (left).

algorithm only depends on the difference between elements within the list, not on their magnitude. We expect this to manifest in the circuits by having lines parallel to the diagonal be fairly uniform.

Head Overlap is introduced to measure the overlap between circuits of two heads h and h'. We take the sum of the absolute difference between elements in the heads normalized by the sum of the absolute value of the elements of the two matrices:

$$\mathcal{O}^{c}_{h,h'} = 1 - \frac{\sum_{ij} \left| W^{c,h}_{ij} - W^{c,h'}_{ij} \right|}{\sum_{ij} \left| W^{c,h}_{ij} \right| + \left| W^{c,h'}_{ij} \right|} \, .$$

The overlap shown in the plots are the mean overlaps of all OV-OV and QK-QK circuit combinations:

$$\mathcal{O} = \frac{1}{2} \frac{1}{N_h} \left(\sum_{c \in \text{OV,QK}} \sum_{h'=1}^{N_h} \sum_{h \neq h'}^{N_h} \mathcal{O}^{c}_{h,h'} \right)$$

C Varying the Model Architecture and Training

In this subsection, we study the impact of varying the model architecture and training such as the number of attention heads, and the use of LN and WD.

C.1 1-Head Model

Fig. 3 shows a single head transformer trained on our list-sorting task. As has been previously noted by Bagiński and Kolly [2023], a single head suffices for list sorting, and the model reaches 100% accuracy at step 133 (left panel of Fig. 3). After the model reaches 100% accuracy, the LLC keeps rising until step 672, with the circuit heat maps looking slightly less noisy and the translational symmetry rising strongly. Thereafter, the LLC drops together with The Circuit Rank as the off-diagonal stripes start to form at step 1985. These stripes become more pronounced towards the end of training (right panel of Fig. 3), but that doesn't seem to have an impact on the LLC. The OV and QK circuits seem to qualitatively function the same in all the panels. This is not surprising, as this model can't undergo head specialization.

The LLC has been calculated with inverse temperature $n\beta = 512/\ln 512 \approx 82$, step size $\epsilon = 10^{-4}$, localization term $\gamma = 32$, $n_{\text{chains}} = 4$ and $n_{\text{draws}} = n_{\text{burnin}} = 2000$.

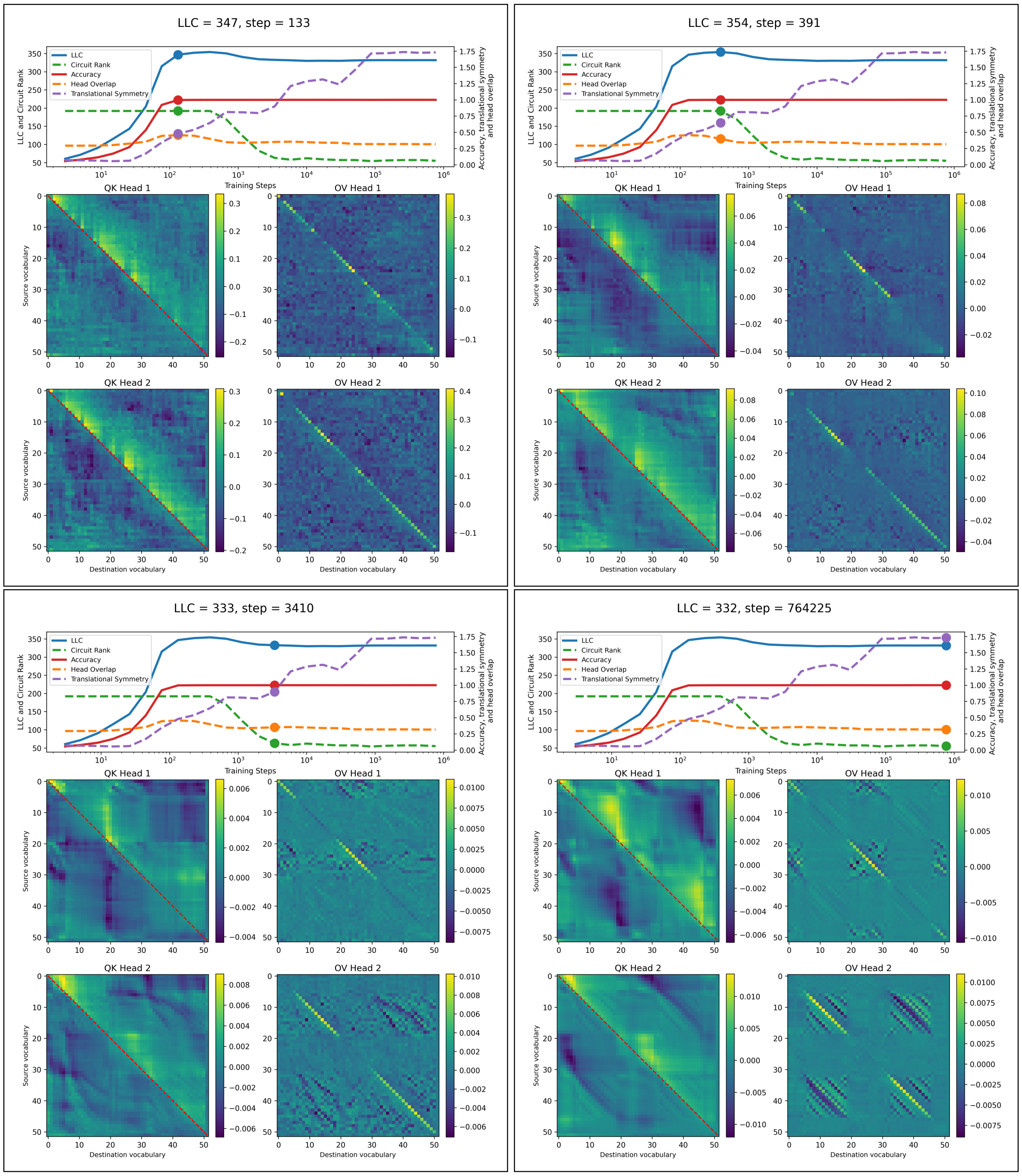

Figure 4: **Baseline 2-head model** as the model reaches 100% accuracy at step 133 (top left panel), peak LLC at step 391 (top right panel), after the LLC has dropped at step 3410 (bottom left), and at the end of training at step 764225 (lower right).

C.2 Baseline 2-Head Model

In Fig. 4 we show the intermediate steps for the evolution of the baseline 2-head model. As previously mentioned, the heads **learns to sort early in the training** at step 133 (upper left of Fig. 4. A few hundred training steps later, the **LLC peaks and vocabulary-splitting begins to emerge** in the OV circuit, although the head overlap is still quite large and the QK circuit doesn't show any distinct partitioning yet.

Next, the **LLC begins to drop while the heads clearly specialize into vocabulary-splitting**, with increasingly non-overlapping regions. We show a snapshot at step 3410 (lower left in Fig. 4), as the LLC stabilizes. The OV circuits of both heads are copying distinct regions. Accordingly, the QK circuit has also developed differentiated patterns along these regions. We expect and observe the attention pattern along dominant destination vocabulary regions to smoothly decrease from left to right, above the diagonal.

These features grow more distinct at the end of training, characterized by the appearance of off-diagonal patterns in the OV circuits. These begin forming after the LLC decrease and finish crystallizing after around 87k training steps. This is well captured by the translational symmetry measure.

C.3 3-Head Model

As shown in the first row of Fig. 5, the 3-head model learns to sort at 100% accuracy at step 133, where all heads attend to and copy overlapping vocabulary regions. At peak LLC (2nd row of Fig. 5) we see first signs of vocabulary-splitting head specialization. As the LLC drops, the overlap between their vocabulary regions decreases, resulting in continguous regions split across three heads, with head 3 covering only a small region (3rd row of Fig. 5). The QK circuits also display differentiated patterns, which upon closer inspection match the active vocabulary regions of the OV circuits. So far, the developmental stages of this model, match those of the baseline 2-head model.

As the evolution continues, around training step 5859 (not shown) the OV circuit of head 3 specializes into an "anti-state", seemingly suppressing the contributions from the other two heads, which behave like in the baseline 2-head model. We identify the state of head 3 to be copy-suppression, as discussed by McDougall et al. [2023]. As the transition occurs, the QK circuit of head 3 also switches to uniform diagonal patterns, not differentiating any vocabulary regions anymore. This transition is not captured by any of our measures. This specialization can be seen at the end of training (4th row of Fig. 5).

The LLC has been calculated with inverse temperature $n\beta = 512/\ln 512 \approx 82$, step size $\epsilon = 10^{-4}$, localization term $\gamma = 32$, $n_{\text{chains}} = 4$ and $n_{\text{draws}} = n_{\text{burnin}} = 60000$.

C.4 4-Head Model

Similar to the other models, the 4-head model also learns to sort with 100% accuracy at step 133 (1st row of Fig. 6). As the LLC decreases, heads begin to specialize with concurrent vocabulary-splitting and copy-suppression appearing in heads 1,3,4 and head 2 respectively (2nd row of Fig. 6). The vocabulary regions are split unevenly, with head 4 covering only a very small region of the vocabulary.

This changes later in the training, after around 87k training steps (3rd row of Fig. 6), with heads 3 and 4 now copying similar vocabulary regions and displaying differentiated attention patterns in the QK circuits. Another transition is visible after 150k training steps (4th row of Fig. 6), where head 3 grows to attend and copy the entire vocabulary range. It seems to be supressing the copy-supression in head 3. This last transition is captured by a small drop in the LLC.

The model remains largely unchanged after this point, until the end of training (5th row of Fig. 6), as is seen from the measures remaining fairly constant.

The LLC has been calculated with inverse temperature $n\beta = 512/\ln 512 \approx 82$, step size $\epsilon = 10^{-4}$, localization term $\gamma = 32$, $n_{\text{chains}} = 4$ and $n_{\text{draws}} = n_{\text{burnin}} = 2000$.

C.5 Baseline 2-head Model without LN

Removing LN from the baseline 2-head model causes a dramatic change to the training dynamics, as shown in Fig. 7. Early in training, at steps 71-348 (top row) the model goes through a transition in

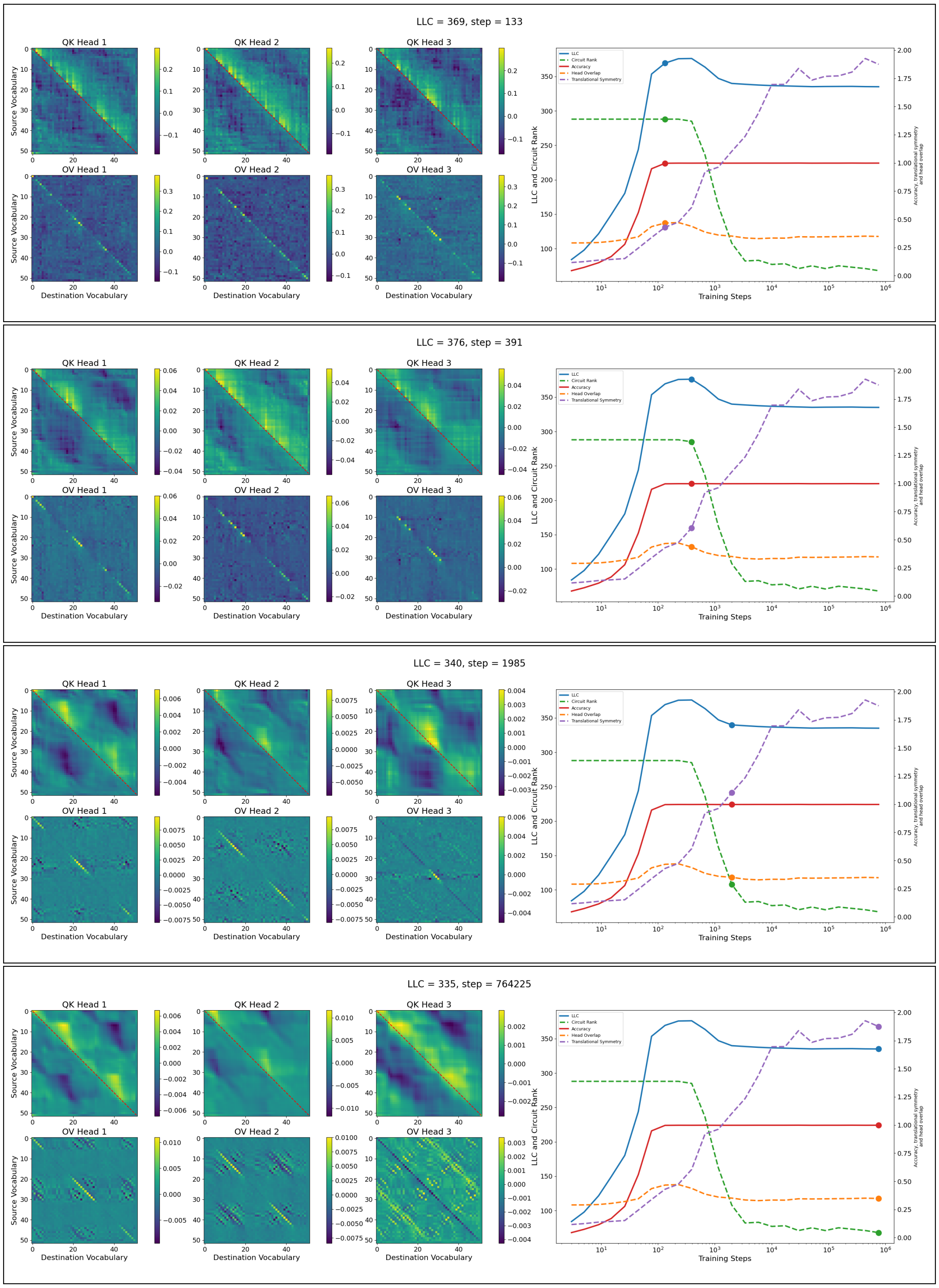

Figure 5: **3-head model** as the model learns how to sort (1st row), at LLC peak (2nd row), three-way vocabulary-splitting after LLC decrease (3rd row) and head 3 performing copy-suppression (4th row).

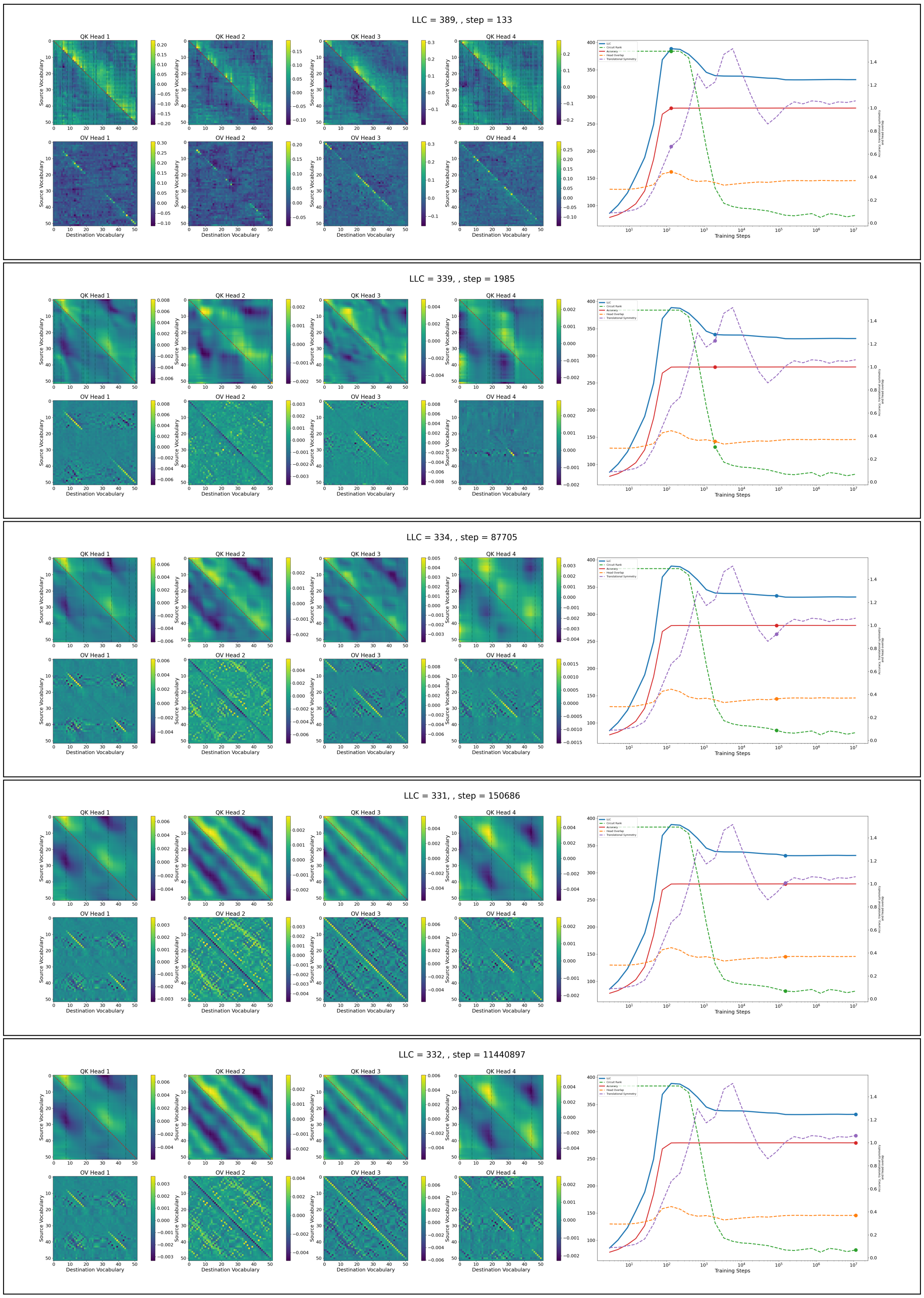

Figure 6: **4-head model** as the model learns how to sort (1st row), as the LLC decreases and heads specialize differently (2nd row), as heads 3 and 4 cover the same vocabulary regions (3rd row), as head 3 covers the entire range (4th row), and at the end of training (5th row).

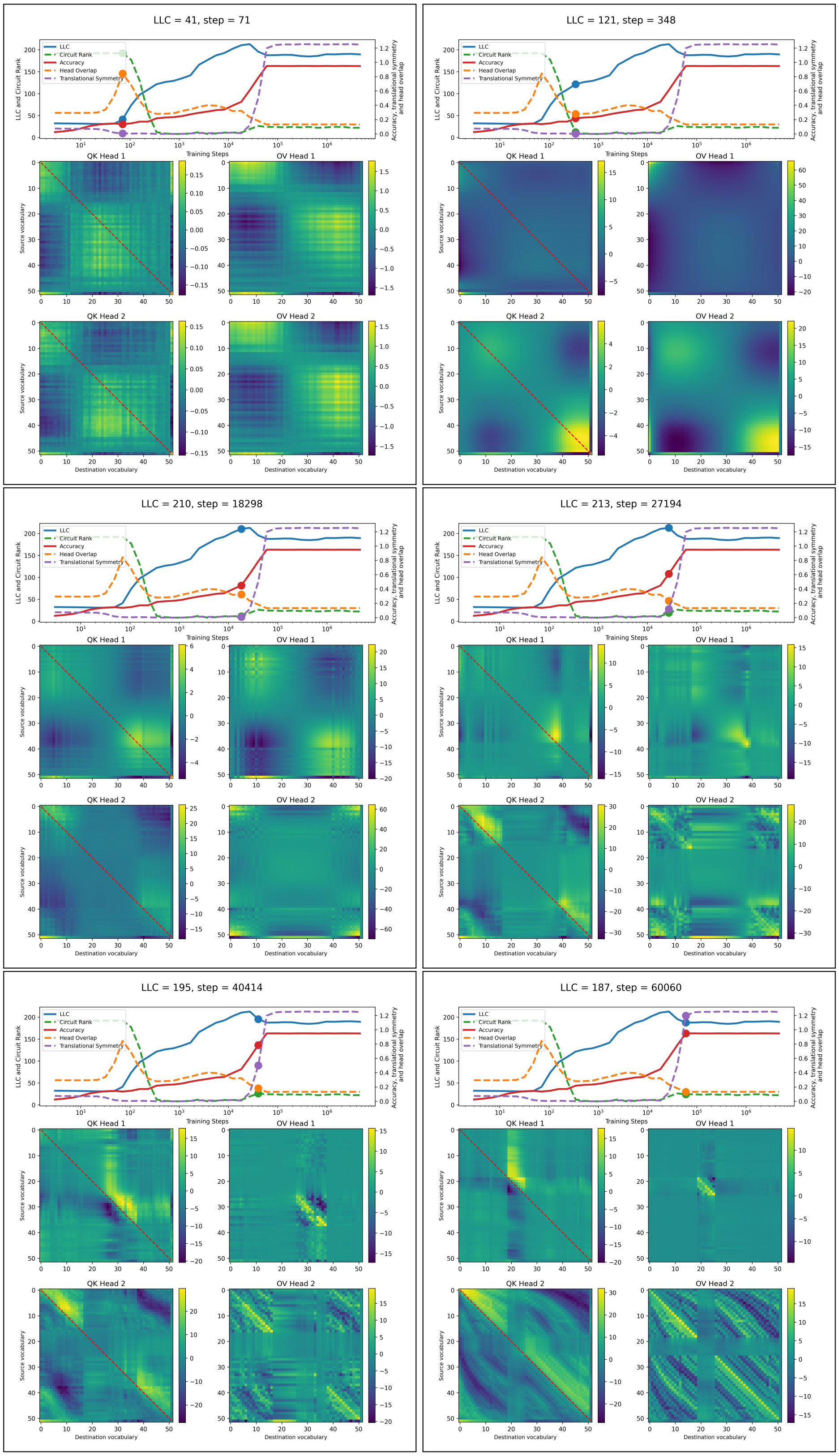

Figure 7: **Baseline 2-head model trained without LN** as the model simplifies but performs poorly (1st row), as relevant structure develops and performance improves rapidly (2nd row), as vocabulary-splitting appears before and after LLC decrease (3rd row).

which The Circuit Rank drops dramatically, the LLC has a sharp increase and the head overlap drops. During this transition, the circuits of the model form a very regular dipole-like pattern.

This dipole-like pattern starts breaking at steps 18298-27194 (middle row) as the LLC peaks, with a formation of the stripe-like patterns parallel to the diagonal in the OV circuit. The QK circuits cover the regions determined by the OV circuit, similar to what we have seen in the other models. This structure formation stabilizes as the LLC drops (bottom row), which is also tracked by a dramatic increase in the translational symmetry. The model never reaches 100% accuracy on list sorting, and the accuracy does not flat-line until step 60060, after which all the measures are stable. The LLC seems to capture the development of this model very well.

The LLC has been calculated with inverse temperature $n\beta = 26$, step size $\epsilon = 10^{-6}$, localization term $\gamma = 32$, $n_{\text{chains}} = 3$ and $n_{\text{draws}} = n_{\text{burnin}} = 100000$.

C.6 Baseline 2-head Model without WD

As seen in Fig. 8, the model without WD still learns to sort at 100% at step 133. The OV and QK circuits seem more noisy, and there is no drop in the Circuit Rank. The LLC still has a large drop between steps 1985 and 10066 during which the heads specialize into splitting the vocabulary. This specialization is clearer for tokens smaller than 20 in the QK and OV circuits, less so for larger vocabulary tokens.

The LLC has been calculated with inverse temperature $n\beta = 512/\ln 512 \approx 82$, step size $\epsilon = 3 \times 10^{-6}$, localization term $\gamma = 56$, $n_{\text{chains}} = 4$ and $n_{\text{draws}} = n_{\text{burnin}} = 65000$.

C.7 Baseline 2-head Model without LN and WD

Fig. 9 shows the evolution of our measures and the circuits for the baseline 2-head model without both LN and WD. Compared to the baseline model, it learns to sort at 100% accuracy somewhat later, at step 391, but considerably faster than the baseline model without LN. The model seems to go via dipole-like circuits around step 45 as the head overlap peaks, very similar to step 71 of the baseline model without LN (compare the top left panels of Figs. 7 and 9). Instead of going via the low Circuit Rank dipole phase, however, the model instead develops circuits that are capable of sorting, while still retaining some of the dipole-like patterns at step 391. This happens at the same time as the LLC peaks.

After this, the LLC drops, and the dipole like pattern gives way to patterns resembling the baseline 2-head model, with partial vocabulary-splitting head specialization in both QK and OV for vocabulary below around 20. We speculate that the reason why the presence of WD causes a worse performance is that it pushes the circuits into simpler low-rank dipole-like patterns instead of learning to sort.

The LLC has been calculated with inverse temperature $n\beta = 30$, step size $\epsilon = 10^{-6}$, localization term $\gamma = 56$, $n_{\text{chains}} = 4$ and $n_{\text{draws}} = n_{\text{burnin}} = 40000$.

D Varying the dataset

In this subsection, we study the impact of varying aspects of the training data, such as the size of the vocabulary, the length of the list and the presence of perturbations in the data set.

D.1 Baseline 2-head Model with Vocabulary Size Increased to 202

Increasing the vocabulary size to 202 produces the training dynamics shown in Fig. 10. The model reaches 97% accuracy around step 635, which coincides with the LLC peak. Thereafter, the LLC and the Circuit Rank drop as the heads specialize into vocabulary-splitting. At the end of training, the model develops a square-like pattern in the QK circuit, which doesn't always correspond to a vocabulary region boundary. This last transition is accompanied by a small drop in The Circuit Rank, but no drop in the LLC.

The LLC has been calculated with inverse temperature $n\beta = 512/\ln 512 \approx 82$, step size $\epsilon = 10^{-3}$, localization term $\gamma = 32$, $n_{\text{chains}} = 4$ and $n_{\text{draws}} = n_{\text{burnin}} = 2000$.

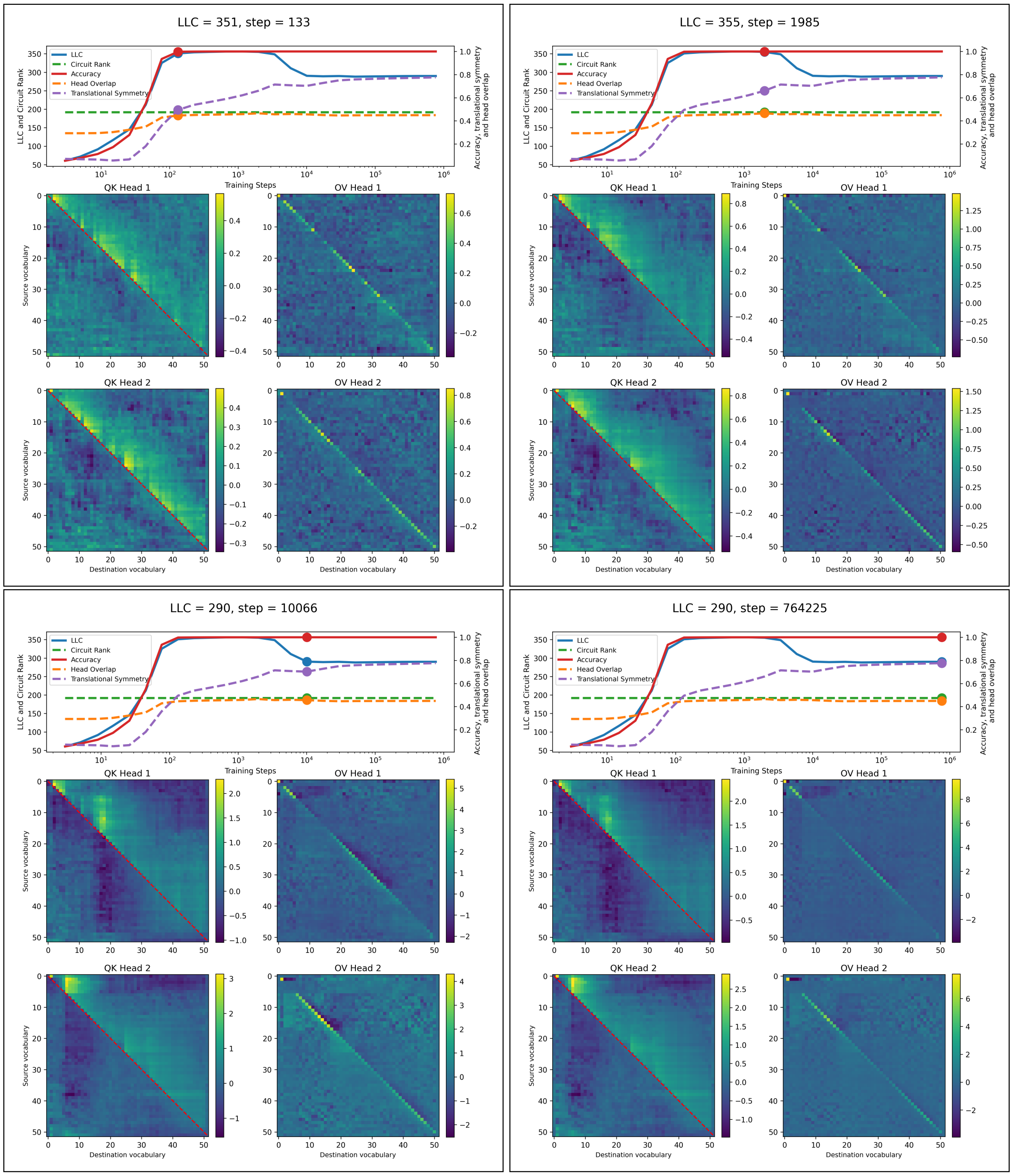

Figure 8: **Baseline 2-head model trained without WD**, as the model learns how to sort (upper left), as the LLC is at its peak (upper right), after the LLC drop (lower left) and at the end of training (lower right).

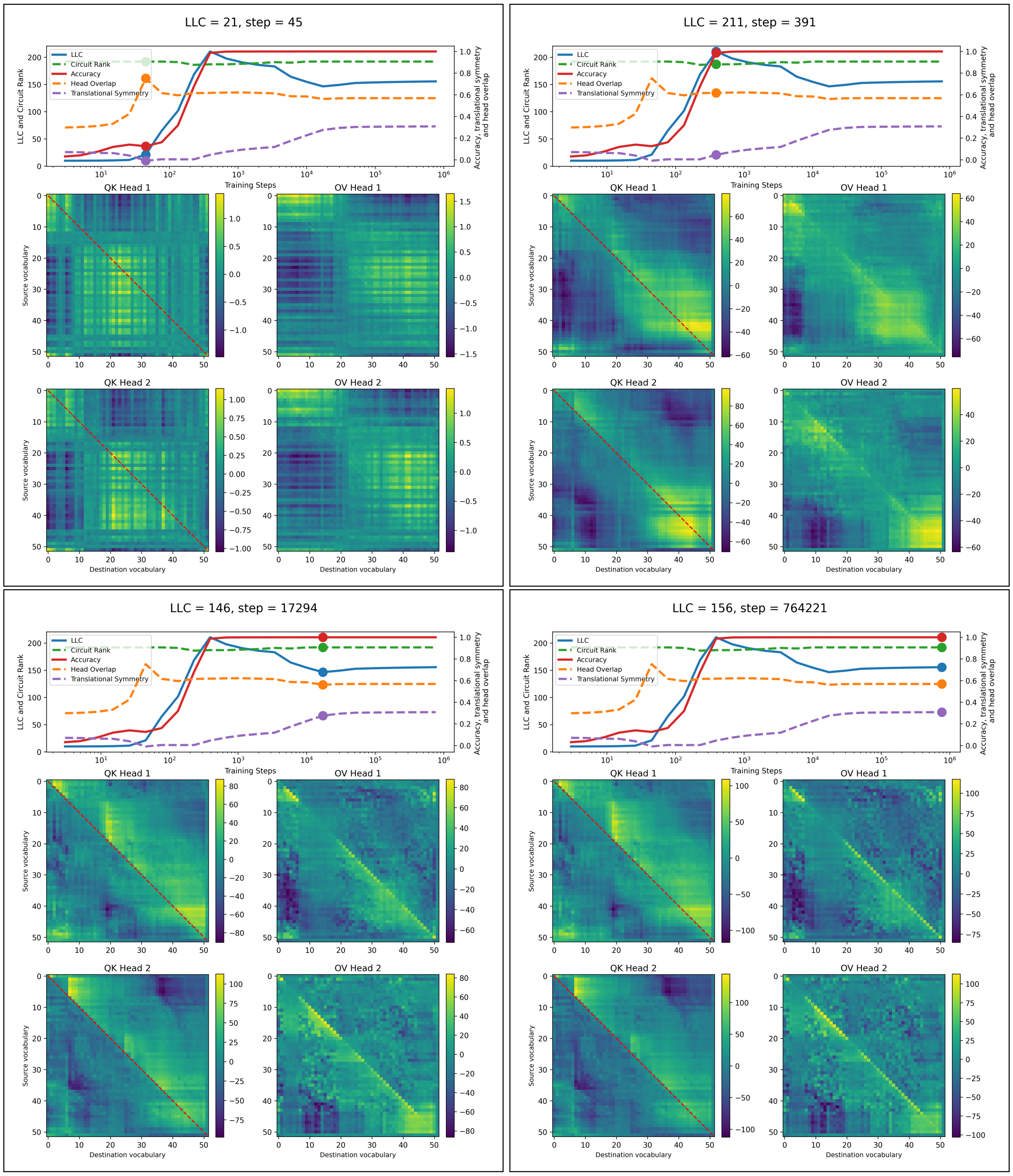

Figure 9: **Baseline 2-head model without LN and WD** as head overlap peaks (upper left), as accuracy is high and LLC peaks (upper right), after LLC drop (lower left) and at the end of training (lower right).

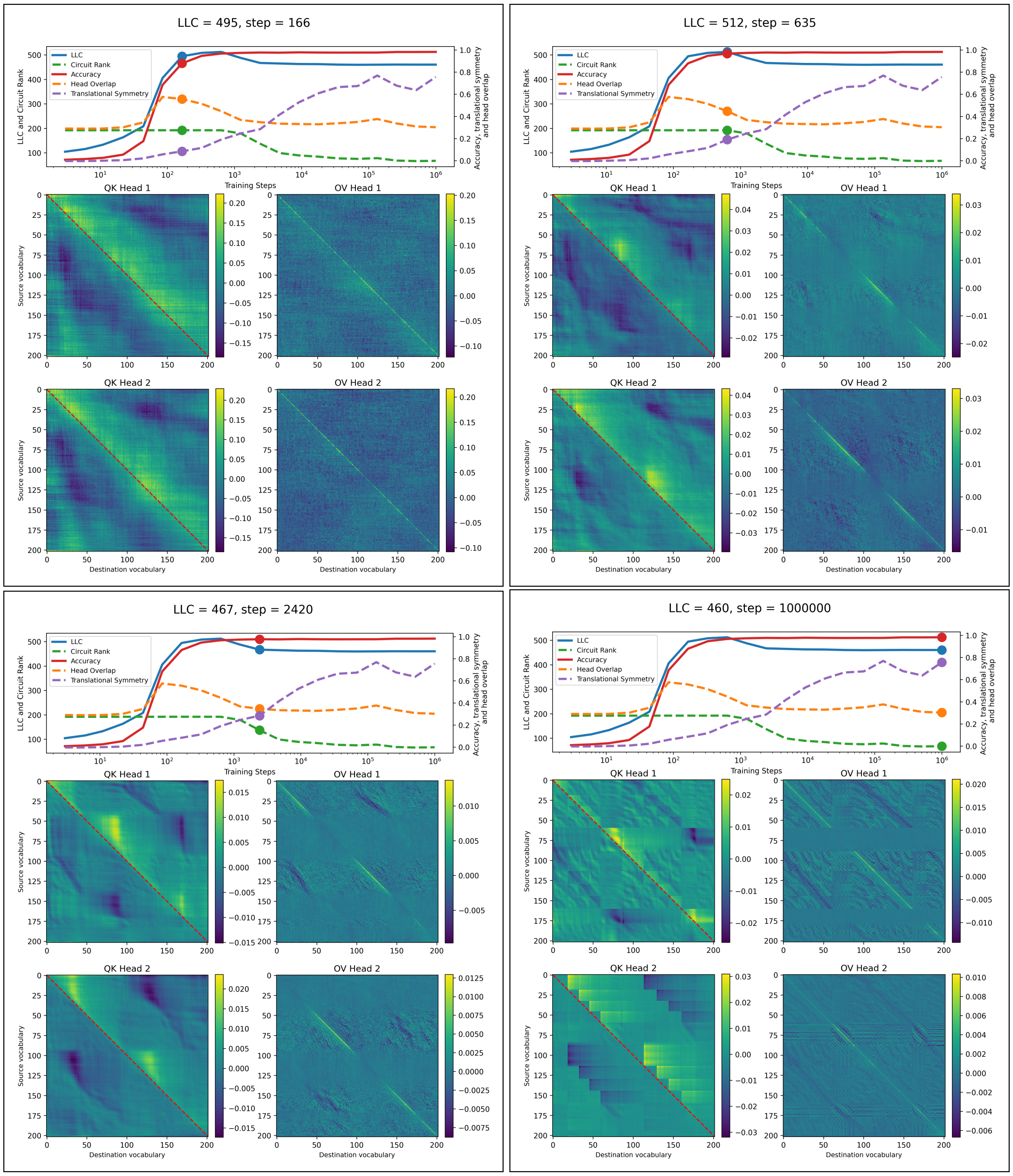

Figure 10: Baseline 2-head model with **vocabulary size increased** to 202, we find similar developmental stages as in the baseline model. **Vocabulary region size increases**.

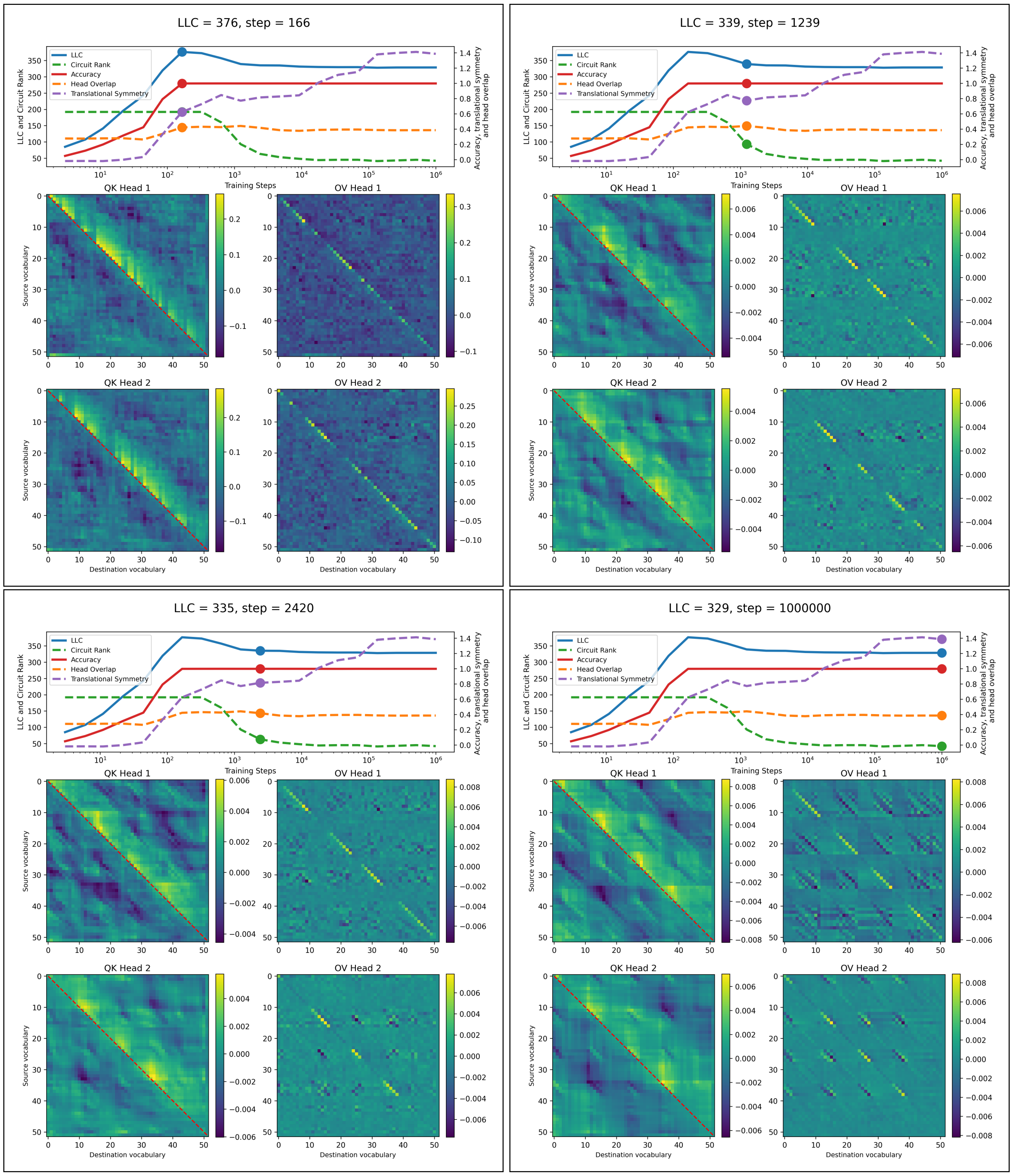

Figure 11: Baseline 2-head model with **list length increased to 20**, we find similar developmental stages as in the baseline model. **Number of vocabulary regions increases.**

D.2 Baseline 2-head Model with List Length Increased to 20

Increasing the list length to 20 yields the training dynamics shown in Fig. 11. Here, the LLC peaks as the model reaches 100% accuracy at step 166, and then drops as the heads specialize into contiguous regions of parameter space and The Circuit Rank drops. We note that compared to the baseline 2-head model, the larger list length leads to a larger number of regions, distributed in a periodic pattern.

The LLC has been calculated with inverse temperature $n\beta = 512/\ln 512 \approx 82$, step size $\epsilon = 3 \times 10^{-6}$, localization term $\gamma = 32$, $n_{\text{chains}} = 4$ and $n_{\text{draws}} = n_{\text{burnin}} = 70000$.

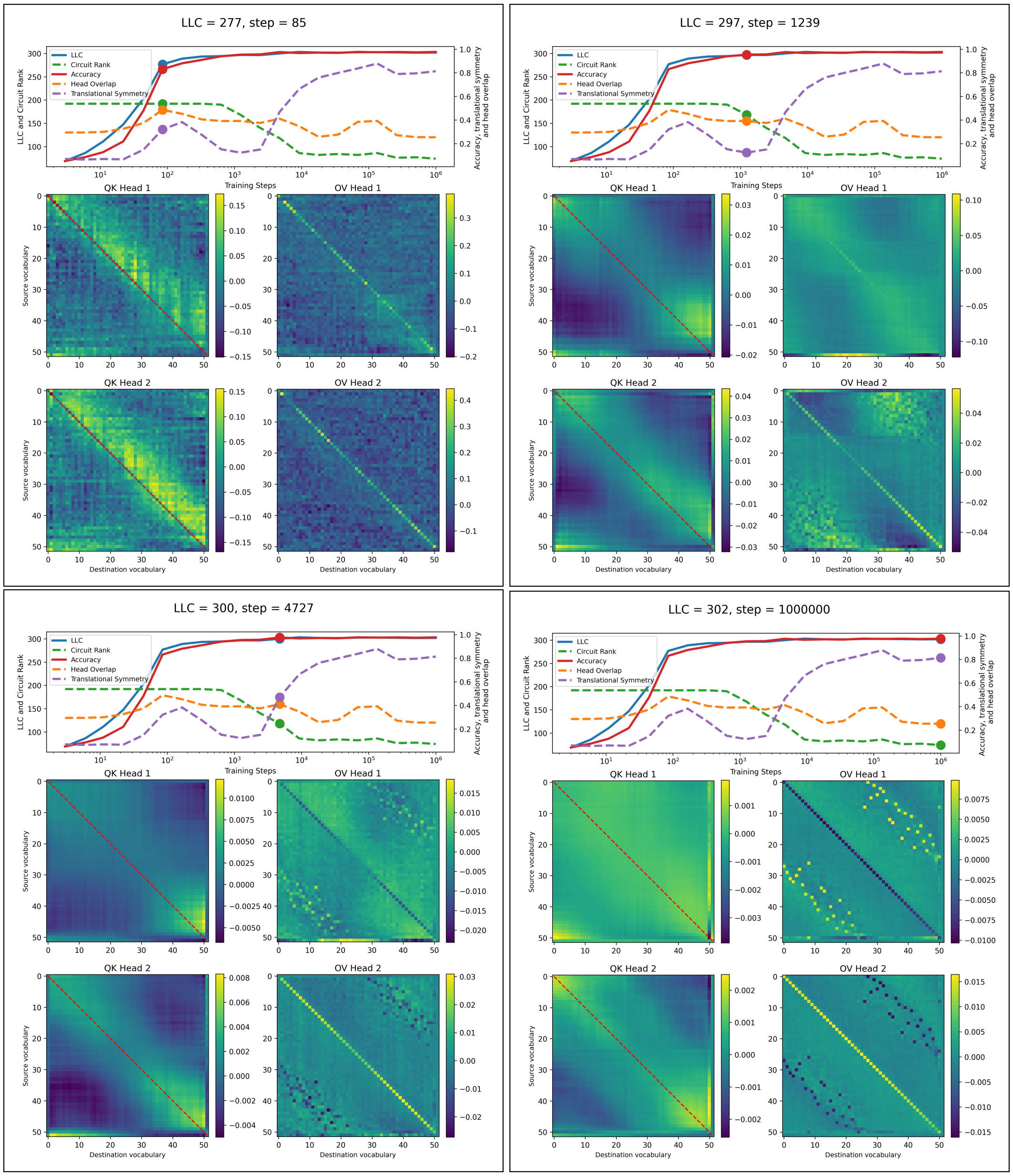

Figure 12: Baseline 2-head model with **perturbed training dataset** shows different developmental stages and it is the only 2-head model where we observe **copy-supression.**

D.3 Baseline 2-head Model with Perturbed Dataset

We perturb the data by iterating through the dataset once, and swapping neighboring elements in the sorted list with probability $40\%/(n_{i+1} - n_i)$, where n_i is the value of the list element i. Since the probability of neighboring elements swapping is always less than 50%, we believe that the optimal strategy still should be to sort the list ignoring the perturbations. The perturbations do, however, have a severe impact on the training dynamics, as shown in Fig. 12.

We don't observe any drop in the LLC, even though The Circuit Rank does drop. The heads don't specialize into vocabulary-splitting modes, but the OV circuits rather settle into what looks like opposites of each other. It looks like head 1 does copy suppression and head 2 does copying, whereas the QK circuits behave very differently from what we have seen in the other models.

The accuracy has been computed on non-perturbed data, and increases throughout training, reaching 98% at the end of training.

The LLC has been calculated with inverse temperature $n\beta = 512/\ln 512 \approx 82$, step size $\epsilon = 10^{-6}$, localization term $\gamma = 32$, $n_{\text{chains}} = 4$ and $n_{\text{draws}} = n_{\text{burnin}} = 200000$.

E Compute

For the experiments, we rented RTX-4090s on vast.ai. We spent a total of \$200, giving about 650 GPU hours. Some experiments had to be re-run and some were not used, and we estimate that about 70% of the compute went into the results included in the paper. The cost of the LLC estimation is proportional to

$$n = n_{\text{chains}} \times (n_{\text{draws}} + n_{\text{burnin}}) = 2n_{\text{chains}} \times n_{\text{draws}}$$

For each model, the LLC has been computed at 24 snapshots, and for each model parameter sweeps have a total cost of approximately $n \approx 2.6\text{M}$. Adding LLC cost of all the experiments used in the paper together we get $n \approx 131\text{M}$. Of the compute used in the paper, we estimate that we spent around 30% on training the models, and 70% on LLC estimation. The computation cost of an experiment, including model training, LLC hyperparameter scan and LLC estimation can be estimated as

$$\left(0.03 + 0.7 \times \frac{2.6\text{M} + 24 \times n_{\text{chains}} \times (n_{\text{draws}} + n_{\text{burnin}})}{131\text{M}}\right) \times 0.7 \times 650\,\text{GPU hours}\,,$$

where n_{chains}, n_{draws} and n_{burnin} is stated for every experiment.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: In sections 4 and 5 we cover the contributions stated in the abstract and introduction.

   Guidelines:

   - The answer NA 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 NA 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: Limitations are covered in section 7

Guidelines:

- The answer NA 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: [NA]

   Justification: We don't introduce new theoretical results.

   Guidelines:

   - The answer NA 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: We cover details of our experimental setups in sections 2.1 and A. LLC hyper-parameters of the experiments covered in the Appendix are stated clearly in the Appendix.

   Guidelines:

- The answer NA 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: [No]

   Justification: The code has not been made open source, but all details required to reproduce the results has been provided.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Hyper-parameters are clearly stated in 2.1, A and in the Appendix. Their choices are discussed in section 7.

   Guidelines:

   - The answer NA 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: [No]

   Justification: This is discussed in the Limitations section, 7.

   Guidelines:

   - The answer NA 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: The machinery used and estimated computation cost is provided in appendix E.

   Guidelines:

   - The answer NA 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: The experiments described in this paper is not of a nature of which there are ethical concerns.

   Guidelines:

   - The answer NA 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: [NA]

    Justification: The work contributes to theoretical understanding of learning dynamics in transformers, and does not have a direct impact on society.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risk.

Guidelines:

- The answer NA 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: We do cite the external software used, and respect licenses.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA 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: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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.