Loss in the Crowd: Hidden Breakthroughs in Language Model Training

Sara Kangaslahti [1] **Elan Rosenfeld** [2] **Naomi Saphra** [1]

Abstract

The training loss curves of a neural network are typically smooth. Any visible discontinuities draw attention as discrete conceptual breakthroughs, while the rest of training is less carefully studied. In this work we hypothesize that similar breakthroughs actually occur frequently throughout training, though their presence is obscured when monitoring the aggregate train loss. To find these hidden transitions, we introduce POLCA, a method for decomposing changes in loss along an arbitrary basis of the low rank training subspace. We use our method to identify clusters of samples that exhibit similar changes in loss through training, disaggregating the overall loss into that of smaller groups of conceptually similar datapoints. We validate our method on synthetic arithmetic, showing that POLCA recovers clusters which represent easily interpretable breakthroughs in the model's capabilities whose existence would otherwise be lost in the crowd.

1. Introduction

As machine learning researchers continue to observe and highlight previously undiscovered phase transitions in training, the community has responded by attempting to characterize the structures and mechanisms that develop during such significant moments. These sudden drops in loss reveal the formation of induction heads (Olsson et al., 2022), syntactic attention structure (Chen et al., 2024a), hierarchical bias (Murty et al., 2023), and many other conceptual breakthroughs (McGrath et al., 2022; Lovering et al., 2022; Power et al., 2022; Abbe et al., 2021). However, the loss curve as a whole remains stubbornly smooth. Phase transitions and momentary concept learning are therefore treated as isolated curiosities; the vast majority of training time is seen as predictable. We will show that in fact, the model is undergoing abrupt conceptual breakthroughs that are concealed by aggregating all data into a single loss curve.

We decompose the loss in two different ways to find hidden breakthroughs. First, we decompose the aggregate loss into loss over individual examples or homogeneous subsets of data. By clustering the loss curves of individual examples, we identify subsets of data that experience synchronized changes in loss stability, implying that they rely on the same conceptual breakthroughs. However, any individual example might benefit from multiple conceptual breakthroughs; in such cases, the example may undergo multiple changes that are synchronized with different subsets of the data. In order to disentangle these breakthroughs, we must instead find different mechanisms or internal changes that affect the loss curve for a given example.

Because we need to disentangle multiple relevant concepts, we introduce a second decomposition, which transforms changes in loss into a collection of responses to movement in specific directions during training. By analyzing these loss curves along specific bases, we identify conceptual breakthroughs that rely on a particular direction of movement. The latter analysis permits further granularity in clustering data, as final performance on an individual example may rely on multiple conceptual breakthroughs, each corresponding to a particular linear direction in training.

- By clustering datapoints on the basis of loss changes during training, we discover that concepts are learned at specific **breakthrough** times. Using changes in datapoint loss to measure stability, we show that smooth aggregated loss curves can conceal momentary inflections in datapoint loss, a scenario we describe as **breakthrough elision**.
- We introduce a modified form of Loss Change Allocation (Lan et al., 2020) called Projection Oriented Loss Change Allocation (POLCA) to measure changes in loss due to parameter adjustments in arbitrary directions during training. Using POLCA, we extend our cluster analysis to identify conceptual breakthroughs that occur in a restricted gradient subspace.

2. Background

What can we learn from transitions in stability? Previous work has extensively documented phase transitions in the stability and sharpness of the loss surface. Jastrzębski

[1] Harvard University [2] Carnegie-Mellon University. Correspondence to: Sara Kangaslahti <sarakangaslahti@g.harvard.edu>.

et al. (2020) point to a clear phase transition in the gradient variance early in training, and Ma et al. (2022) show that such behavior could arise due to the existence of multiple different scales of loss.

Why disaggregate the overall train loss? Individual samples often exhibit changes in loss that are out of line with the monotonic average trend (Xia et al., 2023; Rosenfeld & Risteski, 2024). In full-batch gradient descent, Cohen et al. (2022) identified non-monotonicity arising from oscillation about the maximum Hessian eigenvector. Rosenfeld & Risteski (2024) gave evidence that these oscillations occur across different axes for different samples, and they highlighted human-interpretable semantic features of the data as a likely cause. We hypothesize that movement in these separate directions signals the model's acquisition of distinct capabilities (i.e. "skills" (Arora & Goyal, 2023; Chen et al., 2024b)). To test this hypothesis, and to better identify the semantic meaning of each of these directions, we propose to decompose this instability—defined as the magnitude of oscillation—according to a basis derived from the full loss Hessian at various training checkpoints.

Why decompose the aggregate loss? Similar to the quantization model of parameter scaling of Michaud et al. (2024), we aim to cluster datapoints according to the skills they rely on. However, our POLCA decomposition also addresses what they call *polygenic* scaling effects—samples which combine multiple skills and therefore exhibit breakthroughs at multiple scales. If we assume that a specific skill is enabled by movement along a particular basis vector, then the loss change attributed to movement along the basis vector will stabilize for a sample that requires that skill at the moment the skill is acquired, moving the sample from early to late dynamics through a basis-specific loss phase transition. In other words, by monitoring changes in directions corresponding to specific skills, we support the speculation of Nanda et al. (2023) that *phase transitions are everywhere*.

Why is linear decomposition sufficient? In practice, a conceptual breakthrough would not be expected to occur in a single direction that persists throughout training. However, there is an abundance of evidence that the linear bases of the low rank training subspace (Gur-Ari et al., 2018) are conceptually meaningful. In the late stages of training, linear interpolation between a pair of checkpoints yields a convex path in the loss space (Frankle et al., 2020). Although independently finetuned models with similar generalization heuristics are also linearly connected, interpolations from a nonlinear connection between a model pair with unmatched heuristics fail to generalize with either heuristic (Juneja et al., 2023, ref Appendix D). These observations suggest that linear decomposition should give good results, and our experiments show that the resulting clusters are interpretable in practice.

3. Methods

The key to our approach is the separate consideration of how each individual example's **datapoint loss** changes throughout training. We contrast this individualized metric with the evaluation of in-distribution performance simultaneously across the entire training or validation set, which we call the **aggregated loss**. Using the datapoint loss, we can cluster individual examples on the basis of their loss $L(w_t)$, change in loss $L(w_t) - L(w_{t-1})$, or magnitude of change $|L(w_t) - L(w_{t-1})|$ during training.

3.1. Projection Oriented Loss Change Allocation (POLCA)

Our next objective is to decompose the loss itself into specific directions in the weight space, motivated by several considerations: First, while we have moved from an aggregated loss metric to a more granular datapoint loss metric, we are still only considering breakthroughs that are general enough to be perceived in loss curves. Second, an individual datapoint may benefit from a variety of conceptual breakthroughs, but will not be clustered on the breakthroughs individually. Finally, once we have identified a subset of the data as benefiting from a particular conceptual breakthrough, decomposing into individual weight directions allows us to locate where in the weights the breakthrough occurs and to thereby identify the mechanism involved.

Next we break this loss down by directional movement during training, allowing us to discover breakthroughs that are specific to a given direction. Our procedure, Projection Oriented Loss Change Allocation (POLCA), comprises two steps: first, the selection of the basis, followed by the decomposition of the loss according to that basis.

3.1.1. Finding the basis

Algorithm 1 Finding the POLCA basis

```
input: Training set $X$, Model checkpoints $\{\theta_t\}_{t=1}^{T}$.
$B \leftarrow \emptyset \in \mathbb{R}^{d \times 0}$.
for $t = 1 \ldots T$ do
    $\Pi_\perp \leftarrow I - B(B^\top B)^{-1} B^\top$
    $\mathcal{H} \leftarrow \nabla_\theta^2 \mathcal{L}(X, \theta)$.
    Define $B^+ \in \mathbb{R}^{d \times k}$ as the top $k$ eigenvectors of $\Pi_\perp \mathcal{H}$
    (e.g., via the Lanczos method).
    $B \leftarrow [B, B^+]$.
end for
return $B$
```

We focus on a restricted subspace when decomposing the loss, selecting the basis of this subspace from the maximum eigenvectors of the Hessian matrix. We posit this basis to be interpretable because each basis vector expresses a high gradient covariance and therefore represents a potential decision boundary.

This basis is constructed as follows. Given T intermediate checkpoints throughout training of a model with weights in $\mathbb{R}^d$ and a number k of eigenvectors to compute at each checkpoint, we seek a low rank Tk-dimensional subspace which captures most of the movement during optimization (Gur-Ari et al., 2018). We construct this basis iteratively, starting with $B = \emptyset$: at each checkpoint t we project the model's loss Hessian onto the nullspace of $B \in \mathbb{R}^{d\times(t-1)k}$. We then identify the top k eigenvectors of the resulting projection and append these to B, expanding the dimension. The resulting basis is designed to include directions of highest curvature at each checkpoint so that it will capture synchronized loss behavior throughout training.

3.1.2. Decomposing the Loss

To decompose the loss along our basis, we propose a modified version of Loss Change Allocation (LCA; Lan et al., 2020). LCA is an interpretability tool for analyzing changes in aggregated loss on dataset X between two checkpoints. The output of LCA is the empirical loss change between a pair of checkpoints which can be attributed to the motion of each individual weight unit. Given two consecutive checkpoints with parameters θ_t and θ_{t+1}, LCA reformulates the change in loss as its first-order Taylor approximation, a sum of components which each attribute some loss change to the movement of a single parameter unit $\theta^{(j)}$:

$$\begin{aligned} L(X;\theta_t) &= \sum_{j=0}^{d} (\nabla_\theta L(X;\theta_t))^{(j)} (\theta_{t+1}^{(j)} - \theta_t^{(j)}) && (1) \\ &= \sum_{j=0}^{d} LCA(X;\theta_t^{(j)}) && (2) \end{aligned}$$

The POLCA decomposition differs from LCA in three key ways. First, we do not restrict each direction to correspond to a single unit $\theta^{(j)}$, instead permitting an *arbitrary basis* vector $b \in B$ to replace the axis-aligned basis vectors in LCA; we project onto this basis vector using the dot product $\langle b, \cdot\rangle$. Second, we are interested in changes in the loss on each individual example $x \in X$, not the entire dataset X. These first two modifications provide the following reformulation of LCA.

$$\begin{aligned} L(X;\theta_t) &= \sum_{x\in X} L(x;\theta_t) && (3) \\ &= \sum_{x\in X} \sum_{b\in B} \langle b, \nabla_\theta L(x;\theta_t)\rangle \langle b, \theta_{t+1} - \theta_t\rangle && (4) \end{aligned}$$

The third key difference is that we must use a second order approximation because this basis is constructed explicitly from the Hessian eigenvectors. To understand why this choice of basis requires a second order approximation, recall that each basis vector b is an eigenvector of the Hessian matrix $\mathcal{H}_{t'}(X)$ at some timestep t', where b is chosen because it has the largest eigenvalue $\lambda_{t'}(X,b)$ over the whole dataset. If we assume that the top eigenvectors of the aggregate Hessian maintain high curvature at other points in training and on individual datapoints, then the scaling factor in the second order Taylor term will be very large even at the datapoint level. Limiting the approximation to only the first order term would give a poor estimate, as the second order term could be expected to dominate.

Exact computation of the second order term would be intractable, requiring computation of the top eigenvalues/vectors for each individual datapoint x. Instead, we can approximate it by substituting the true eigenvalue, denoted $\lambda_t(X,b) := b^\top \mathcal{H}_t(X) b$, with the curvature of the individual loss in the direction b, i.e. $\lambda_t(x,b) = b^\top \mathcal{H}_t(x) b$. If the aggregate Hessian eigenvector b is close to the span of the top eigenvectors of the datapoint-specific Hessian for x, this provides a reasonable estimate while reducing calculation to a single hessian-vector product per eigenvector. We therefore approximate the basis projection of the datapoint Hessian $h(x,b,\theta_t)$ as detailed in Appendix A.

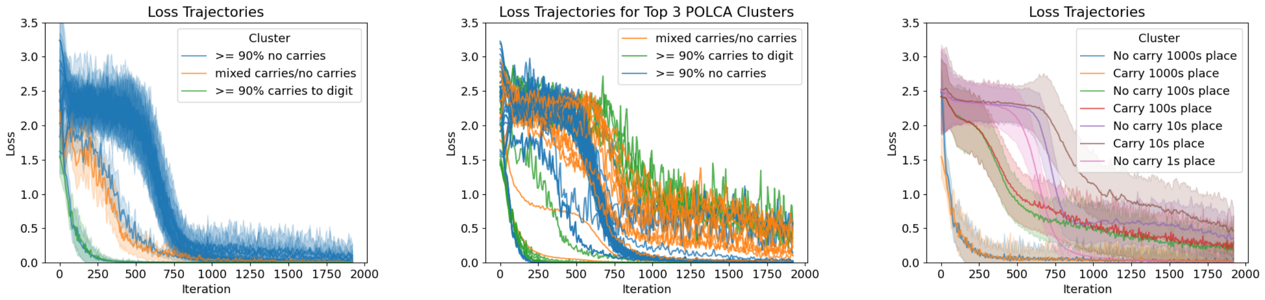
$$\begin{aligned} h(x,b,\theta_t) &= \frac{\lambda_t(x,b)}{2} \langle \theta_{t+1} - \theta_t, b\rangle^2 \\ &\approx \frac{\lambda_t(X,b)}{2} \cdot \langle \theta_{t+1} - \theta_t, b\rangle^2 \\ &\quad \times \frac{\langle L(x;\theta_{t+1}) - L(x;\theta_t), b\rangle}{\langle L(X;\theta_{t+1}) - L(X;\theta_t), b\rangle} && (5) \\ &= \tilde{h}(x,b,\theta_t) \end{aligned}$$

Recall that b is selected to maximize the full dataset eigenvalue $\lambda(X,b)$ at some timestep. Limiting the approximation to only the first order should therefore lead to a poor approximation, as the second order term may dominate the first order. To account for the increase in error, we modify Equation 4 into the second order Taylor expansion using the approximation from Equation 5.

$$\begin{aligned} L(X;\theta_t) \approx & \textstyle\sum_{x\in X} \sum_{b\in B} \langle b, \nabla_\theta L(x;\theta_t)\rangle \langle b, \theta_{t+1} - \theta_t\rangle \\ & + \tilde{h}(x,b,\theta_t) && (6) \\ = & \textstyle\sum_{x\in X} \sum_{b\in B} POLCA(x,b;\theta_t) && (7) \end{aligned}$$

4. Arithmetic language modeling

We find that breakthrough clustering can, in fact, reveal discretely learned concepts and natural kinds within the data, even when those kinds are not discoverable by clustering directly on loss curves.



(a) Loss change magnitude clusters colored by carrying skill.

(b) Top 3 POLCA clusters for each vector colored by carrying skill.

(c) Ground truth mean loss curves clustered by carry and digit skill.

Figure 1. Average cluster loss curves for different breakthrough clustering methods on the skill-it addition dataset, and for the ground truth subsets that correspond to each cluster's dominant set of skills. Using POLCA and visualizing the top 3 clusters per vector, we find clusters corresponding to the carrying skill for various digits (shown in green in (a)), which has a different loss curve to the other skills but is challenging to recover using solely the aggregate loss.

4.1. The data

Our synthetic experiments use data from the arithmetic addition setting in Chen et al. (2024b), where the model is trained to compute the sum of two 3-digit numbers. This setting has 4 skills corresponding to each of the digits in the output sum. We note that the digit in the 1000s place is always a 0 or a 1 since the two numbers being summed have 3 digits. As shown in Appendix Figure 2 and Chen et al. (2024b), the skills corresponding to the digits have different loss curves, so they provide a baseline for how well breakthrough clustering can recover skills with different loss curves. We also consider two additional skills: carries to the output token and the ground truth output value. The output value is a simple skill that is trivial to cluster on using the data, whereas the carries and digits represent skills that we are interested in recovering but are unknown in the real-world setting.

Experimental setup We train a 2-layer transformer model with embedding dimension 512, 4 attention heads, and an MLP dimension of 2048. For a validation set with 1250 data points and 5000 output tokens, we compute the loss and POLCA values for each token at intervals of 5 iterations throughout training. We compute the POLCA basis using the eigenvectors of the Hessian estimated using a 1250 data point sample of the training set as detailed in Algorithm 1. We compute a new basis vector every 100 iterations.

We then analyze breakthrough clustering on the loss and POLCA trajectories in 4.2 and 4.3. We use Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) (Campello et al., 2013), as we are interested in discovering clusters of curves that may have different densities. The HDBSCAN outliers are shown as cluster -1 in Appendix C but are excluded from Figure 1.

4.2. Recovering concepts from the exact loss

In our clustering experiments on arithmetic, we first consider whether decomposition is necessary for identifying individual concepts. To this end, we cluster solely on the magnitude of the change in exact per-token loss for successive timesteps, rather than using the decomposed estimations. As shown in Figure 5, we do find that it is possible to recover, to a substantial degree, the output token value, making it clear that this skill corresponds to example difficulty. However, the clusters are much less homogeneous with respect to the digit and carry skills, especially for the 10s and 100s digits, which have similar loss change magnitude curves (Figure 1a, Appendix Figure 3). We also observe similar results for clustering on the loss (Appendix Figure 4). We will demonstrate a clear improvement in the recovery of complex skills and the interpretability of clusters after POLCA decomposition.

4.3. Recovering concepts with POLCA

Due to the shortcomings of clustering solely on the loss, we instead cluster on the loss changes decomposed by POLCA, separately considering each basis vector. The POLCA value for a given token and basis vector represents the loss change attributed to movement along that vector. We find that certain vectors have homogeneous clusters corresponding to carrying skills, such as vector 4 (Appendix Figure 10) and vector 0 (Appendix Figure 6). The skill homogeneity is high for the majority of vectors and is shown for the first 5 vectors in Appendix C.

Figure 1b shows the trajectories of the three top HDBSCAN clusters for each vector. The top clusters are able to recover different vectors corresponding to the various digit and carry trends depicted in Figure 1c. Furthermore, the top clusters at different vectors can be used to understand which directions are important for learning a specific skill and when these directions emerge as top eigenvectors in the Hessian. As a result, we have shown that breakthrough clustering on the POLCA vectors can be used to find when complex skills are learned and better understand how they are learned.

Acknowledgements

We thank the anonymous reviewers for their feedback. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 2140743. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation. This work was enabled in part by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence.

We used the TransformerLens library (Nanda & Bloom, 2022) for some of the experiments run in this paper.

Author Contributions

Sara Kangaslahti designed and ran the experiments and contributed to paper writing. Elan Rosenfeld co-supervised the project and contributed to the paper writing. Naomi Saphra advised and co-supervised the project and led the paper writing.

References

Abbe, E., Boix-Adsera, E., Brennan, M., Bresler, G., and Nagaraj, D. The staircase property: How hierarchical structure can guide deep learning, 2021.

Arora, S. and Goyal, A. A theory for emergence of complex skills in language models. *arXiv preprint arXiv:2307.15936*, 2023.

Campello, R. J. G. B., Moulavi, D., and Sander, J. Density-based clustering based on hierarchical density estimates. In Pei, J., Tseng, V. S., Cao, L., Motoda, H., and Xu, G. (eds.), *Advances in Knowledge Discovery and Data Mining*, pp. 160–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-37456-2.

Chen, A., Shwartz-Ziv, R., Cho, K., Leavitt, M. L., and Saphra, N. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in mlms, 2024a.

Chen, M., Roberts, N., Bhatia, K., Wang, J., Zhang, C., Sala, F., and Ré, C. Skill-it! a data-driven skills framework for understanding and training language models. *Advances in Neural Information Processing Systems*, 36, 2024b.

Cohen, J. M., Kaur, S., Li, Y., Kolter, J. Z., and Talwalkar, A. Gradient descent on neural networks typically occurs at the edge of stability, 2022.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

Gur-Ari, G., Roberts, D. A., and Dyer, E. Gradient descent happens in a tiny subspace, 2018.

Jastrzębski, S., Szymczak, M., Fort, S., Arpit, D., Tabor, J., Cho*, K., and Geras*, K. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2020.

Juneja, J., Bansal, R., Cho, K., Sedoc, J., and Saphra, N. Linear connectivity reveals generalization strategies. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hY6M0JHl3uL.

Lan, J., Liu, R., Zhou, H., and Yosinski, J. Lca: Loss change allocation for neural network training, 2020.

Lovering, C., Forde, J., Konidaris, G., Pavlick, E., and Littman, M. Evaluation beyond task performance: Analyzing concepts in alphazero in hex. *Advances in Neural Information Processing Systems*, 35:25992–26006, 2022.

Ma, C., Kunin, D., Wu, L., and Ying, L. Beyond the quadratic approximation: the multiscale structure of neural network loss landscapes. *arXiv preprint arXiv:2204.11326*, 2022.

McGrath, T., Kapishnikov, A., Tomašev, N., Pearce, A., Wattenberg, M., Hassabis, D., Kim, B., Paquet, U., and Kramnik, V. Acquisition of chess knowledge in alphazero. *Proceedings of the National Academy of Sciences*, 119 (47):e2206625119, 2022.

Michaud, E. J., Liu, Z., Girit, U., and Tegmark, M. The quantization model of neural scaling, 2024.

Murty, S., Sharma, P., Andreas, J., and Manning, C. D. Grokking of hierarchical structure in vanilla transformers. *arXiv preprint arXiv:2305.18741*, 2023.

Nanda, N. and Bloom, J. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens, 2022.

Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. Progress measures for grokking via mechanistic interpretability, 2023. URL https://arxiv.org/abs/2301.05217.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Power, A., Burda, Y., Edwards, H., Babuschkin, I., and Misra, V. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Rosenfeld, E. and Risteski, A. Outliers with opposing signals have an outsized effect on neural network optimization. In *The Twelfth International Conference on Learning Representations*, 2024.

Xia, M., Artetxe, M., Zhou, C., Lin, X. V., Pasunuru, R., Chen, D., Zettlemoyer, L., and Stoyanov, V. Training trajectories of language models across scales, 2023.

A. Derivation of approximate second order term

$$
\begin{aligned}
g_{t+1}(X) - g_t(X) &\approx \mathcal{H}_t(X)(\theta_{t+1} - \theta_t) && (8)\\
\langle g_{t+1}(X) - g_t(X), b\rangle &\approx b^\top \mathcal{H}_t(X) b \langle b, \theta_{t+1} - \theta_t\rangle && (9)\\
&= \lambda_t(X)\langle b, \theta_{t+1} - \theta_t\rangle && (10)
\end{aligned}
$$

If we assume b to also be an eigenvector of the datapoint Hessians $\mathcal{H}'_t(x)$, we can apply a similar argument on the data point level.

$$
\langle g'_{t+1}(x) - g'_t(x), b\rangle \approx b^\top \mathcal{H}'_t(x) b \langle b, \theta_{t+1} - \theta_t\rangle \quad (11)
$$

Then we may approximate it as:

$$
\begin{aligned}
\frac{g'_{t+1}(x) - g'_t(x)}{g_{t+1}(X) - g_t(X)} &\approx \frac{\mathcal{H}'_t(x)(\theta_{t+1} - \theta_t)}{\mathcal{H}_t(X)(\theta_{t+1} - \theta_t)} && (12)\\
\left\langle \frac{g'_{t+1}(x) - g'_t(x)}{g_{t+1}(X) - g_t(X)}, b \right\rangle &\approx \frac{b^\top \mathcal{H}'_t(x) b \langle b, \theta_{t+1} - \theta_t\rangle}{\lambda_t(X, b)\langle b, \theta_{t+1} - \theta_t\rangle} && (13)\\
\left\langle \frac{g'_{t+1}(x) - g'_t(x)}{g_{t+1}(X) - g_t(X)}, b \right\rangle &\approx \frac{\langle h'_t(x), b\rangle \langle b, \theta_{t+1} - \theta_t\rangle}{\lambda_t(X, b)\langle b, \theta_{t+1} - \theta_t\rangle} && (14)\\
\lambda_t(X, b)\left\langle \frac{g'_{t+1}(x) - g'_t(x)}{g_{t+1}(X) - g_t(X)}, b \right\rangle &\approx \langle h'_t(x), b\rangle && (15)
\end{aligned}
$$

B. Undecomposed trajectories for the digit skill

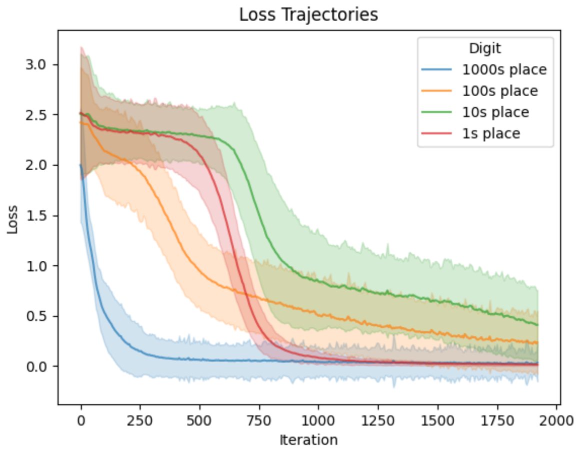


Figure 2. Mean and standard deviation of the loss trajectories for each digit.

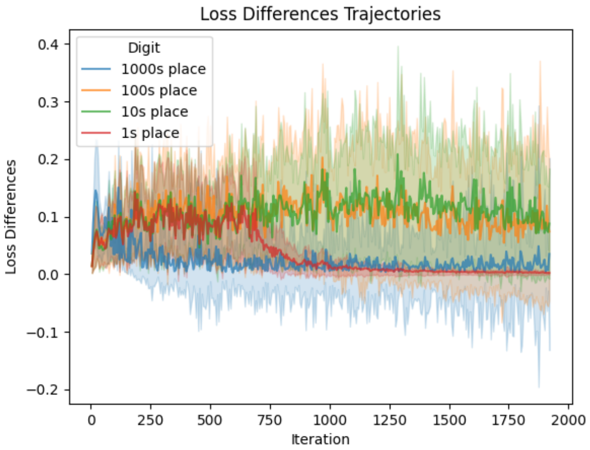
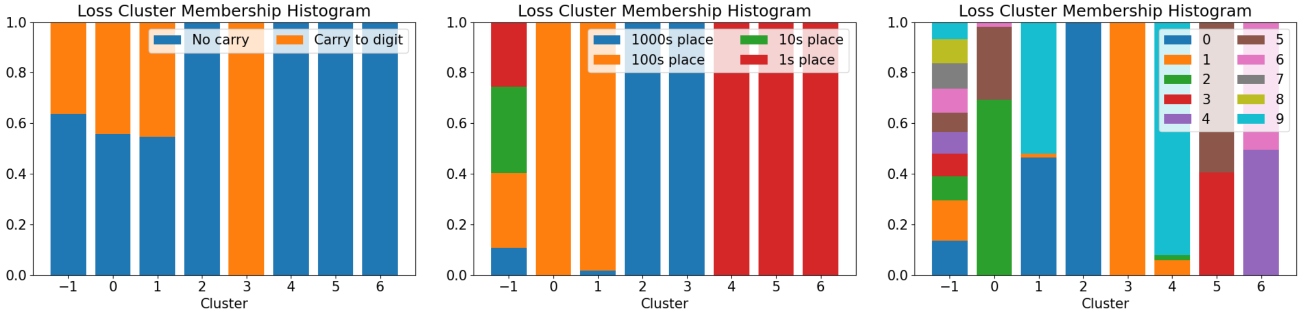


Figure 3. Mean and standard deviation of the loss difference magnitude trajectories for each digit.

C. Additional cluster histograms

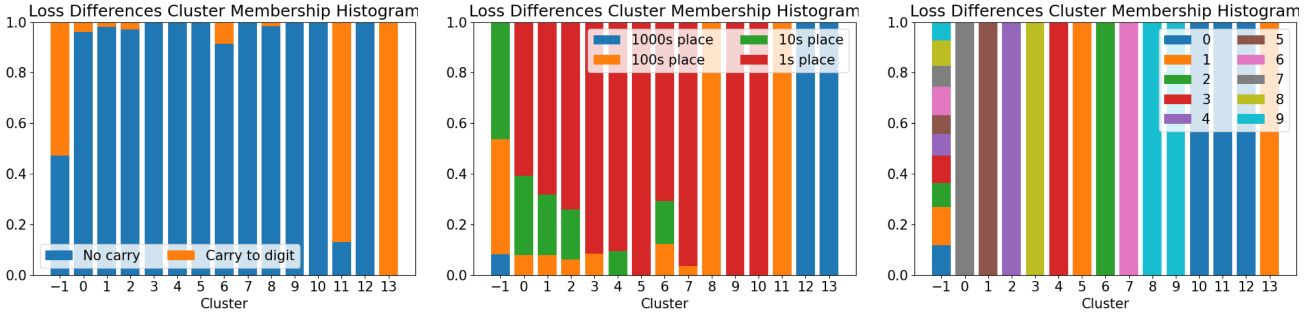


(a) Undecomposed loss carry clusters.

(b) Undecomposed loss digit clusters.

(c) Undecomposed loss output clusters.

Figure 4. Dominant skill homogeneity for breakthrough clustering on the loss trajectories on the skill-it addition dataset. Bars are colored by category in (a) whether there is a carry to the token or not, (b) the digit, and (c) the value of the correct output token. While these clusters are homogeneous with respect to the digit, they do not recover the carrying skill.

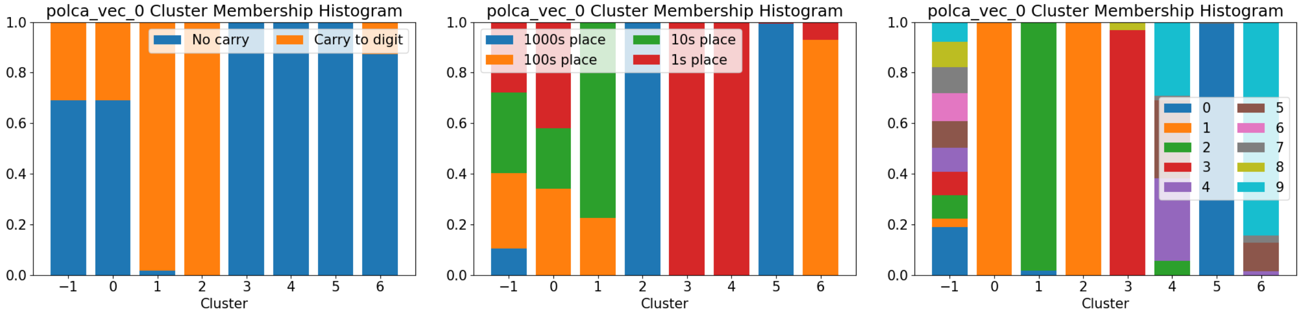


(a) Undecomposed carry clusters.

(b) Undecomposed digit clusters.

(c) Undecomposed output clusters.

Figure 5. Dominant skill homogeneity for breakthrough clustering on the absolute loss difference trajectories on the skill-it addition dataset. Bars are colored by category in (a) whether there is a carry to the token or not, (b) the digit, and (c) the value of the correct output token. While these clusters are homogeneous with respect to the value of the correct output token, they do not fully recover the carrying skill.

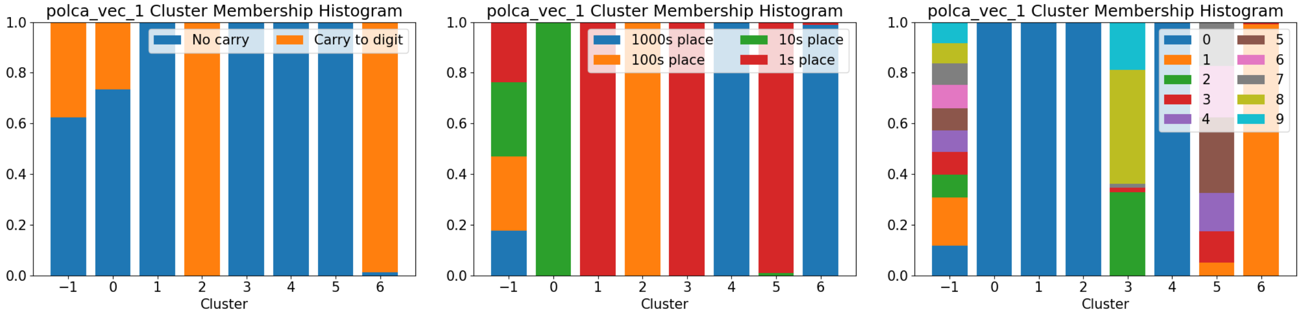


(a) POLCA vector 0 carry clusters.

(b) POLCA vector 0 digit clusters.

(c) POLCA vector 0 output clusters.

Figure 6. Dominant skill homogeneity for breakthrough clustering on the 0th POLCA vector trajectories on the skill-it addition dataset. Bars are colored by (a) whether there is a carry to the token or not, (b) the digit, and (c) the value of the correct output token. These clusters are homogeneous with respect to the carry skill and represent a direction that is important for learning the carrying skill.

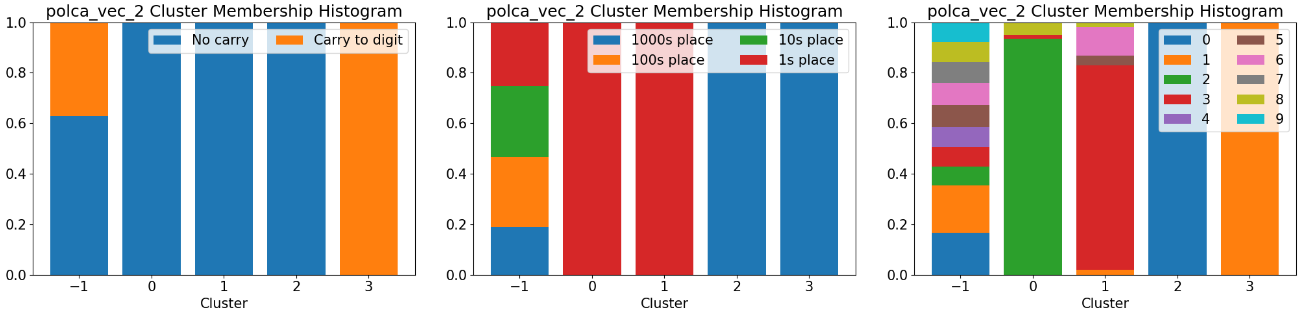


(a) POLCA vector 1 carry clusters.

(b) POLCA vector 1 digit clusters.

(c) POLCA vector 1 output clusters.

Figure 7. Dominant skill homogeneity for breakthrough clustering on the 1st POLCA vector trajectories on the skill-it addition dataset. Bars are colored by (a) whether there is a carry to the token or not, (b) the digit, and (c) the value of the correct output token. These clusters are homogeneous with respect to the carry and digit skills.

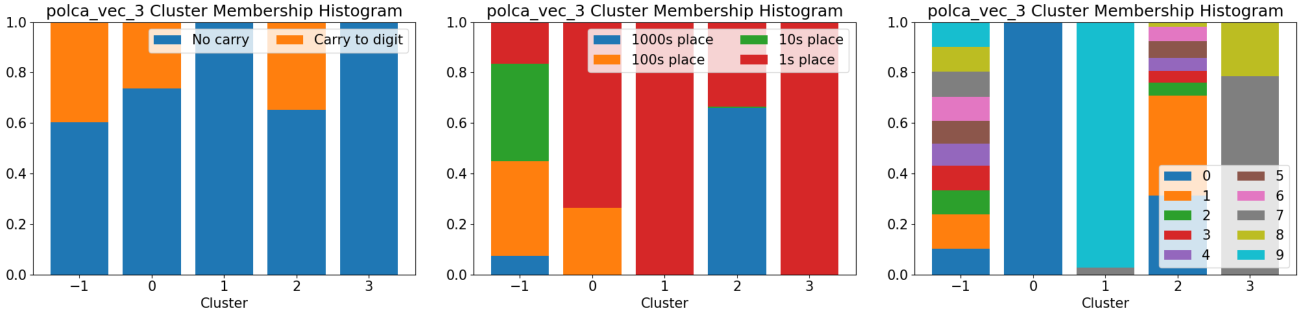


(a) POLCA vector 2 carry clusters.

(b) POLCA vector 2 digit clusters.

(c) POLCA vector 2 output clusters.

Figure 8. Dominant skill homogeneity for breakthrough clustering on the 2nd POLCA vector trajectories on the skill-it addition dataset. Bars are colored by (a) whether there is a carry to the token or not, (b) the digit, and (c) the value of the correct output token. These clusters are homogeneous with respect to the carry and digit skills.

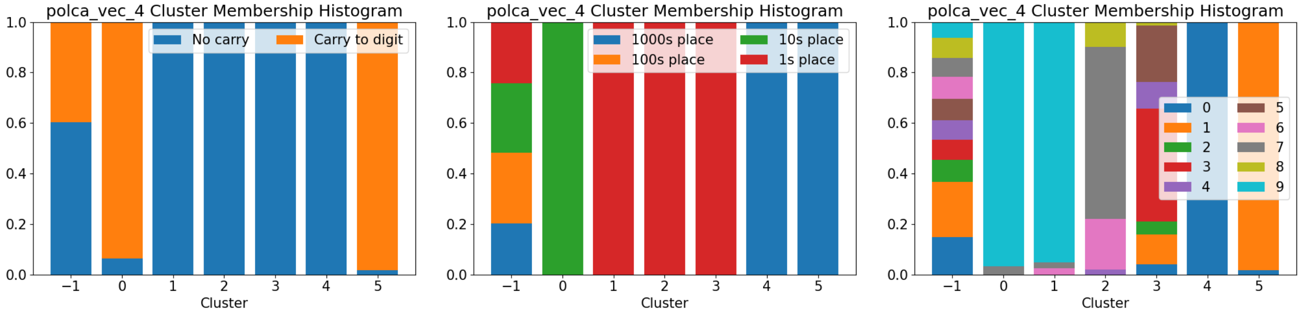


(a) POLCA vector 3 carry clusters.

(b) POLCA vector 3 digit clusters.

(c) POLCA vector 3 output clusters.

Figure 9. Dominant skill homogeneity for breakthrough clustering on the 3rd POLCA vector trajectories on the skill-it addition dataset. Bars are colored by (a) whether there is a carry to the token or not, (b) the digit, and (c) the value of the correct output token. These clusters are fairly homogeneous with respect to the combination of the three types of skills.



(a) POLCA vector 4 carry clusters.

(b) POLCA vector 4 digit clusters.

(c) POLCA vector 4 output clusters.

Figure 10. Dominant skill homogeneity for breakthrough clustering on the 4th POLCA vector trajectories on the skill-it addition dataset. Bars are colored by (a) whether there is a carry to the token or not, (b) the digit, and (c) the value of the correct output token. These clusters are homogeneous with respect to the carry and nominal skills and represent a direction that is important for learning the carrying skill.