

## Appendix A Methods

### A.1 Models

All models used in our study are listed in Table 1. We employ the base versions (i.e., without fine-tuning), since our prompts do not include any instructions in its format, as described in Section A.2.

Table 1: Details on models used in our study, including maximum context length.

Architecture	Version	Size	Context Length
Llama	2	7b	4k
	3.1	8b	132k
	3.2	1b	132k
	3.2	3b	132k
Gemma	2	9b	8k
Mistral	0.1	7b	4k <sup>a</sup>

<sup>a</sup>Original context window of 4k, but extendable to 16k with a sliding window attention (SWA) mechanism.

### A.2 Prompt

We format our prompts by presenting the token pairs  $(x, y)$  as direct concatenations without any separator, punctuation, or instructional context. For instance, if the pair is  $(A, B)$ , the prompt would contain  $AB$  (for  $r = 1$ ) without a space or symbol between them. This minimal setup ensures that the model relies purely on co-occurrence patterns to form associations, rather than leveraging syntactic or structural cues. All models under study include a beginning-of-sequence (BOS) token, which we consistently use as the first token in every prompt. To avoid degenerate token pairs, we restrict the vocabulary space by excluding stop words, punctuation, and numerals.

### A.3 Vocabulary sampling

As detailed in Section 3.3, for each model, we randomly sampled 1,000 tokens from  $\mathcal{V}$  to form the representative subset  $\mathcal{V}$ , resulting in approximately 1M pairwise token combinations.

Figure 4 presents heatmaps showing how the 1 million sampled token pairs are distributed across pairwise similarity before learning (x-axis) and vocabulary interference (y-axis), for each individual model (a–f) and across all models combined (g). The color scale indicates the log-transformed number of token pairs in each bin. These distributions reflect the natural data availability prior to applying uniform sampling of 10 items per bin. Overall, the density of sampled pairs tends to concentrate in the low-to-mid similarity and interference ranges, with some variation across models.

To ensure balanced coverage across the pairwise similarity and vocabulary interference space, we applied a uniform sampling strategy to construct the set  $\mathcal{Q}_m$  for each model. All token pairs were first assigned to bins according to their pair similarity and vocabulary interference values. We then counted how many token pairs already existed in each bin from the original set  $\mathcal{P}_m$ , and filtered out any duplicates to avoid reusing token pairs. The final set  $\mathcal{Q}_m$  was created by combining the original and newly sampled pairs, resulting in an approximately uniform distribution of token pairs across the similarity–interference grid. Figure 5 illustrates the resulting distributions after this sampling procedure. Subfigures (a–f) show the heatmaps for each model individually, while (g) displays the combined heatmap representing the aggregated distribution across all models. Each cell indicates the (log-transformed) number of token pairs in the corresponding pair similarity  $\times$  vocabulary interference bin. As intended, the distributions are largely uniform, with minor deviations due to constraints in available data for certain bins.

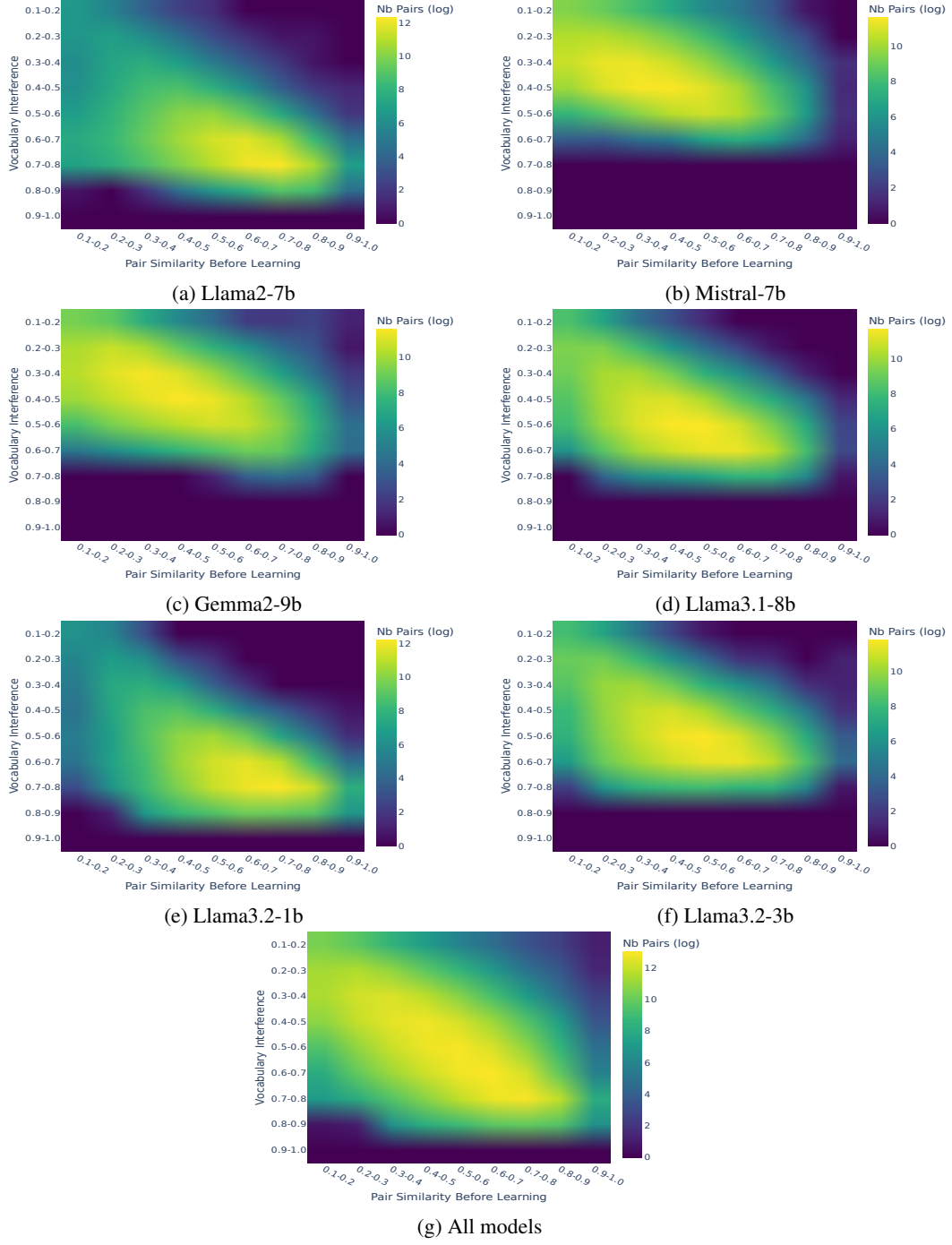


Figure 4: Log-scale heatmap showing the joint distribution of token pairs across pairwise similarity before learning (x-axis) and vocabulary interference (y-axis) in the representative vocabulary subset  $\tilde{\mathcal{V}}$ . Subplots (a–f) correspond to individual models; subplot (g) aggregates results across all models.

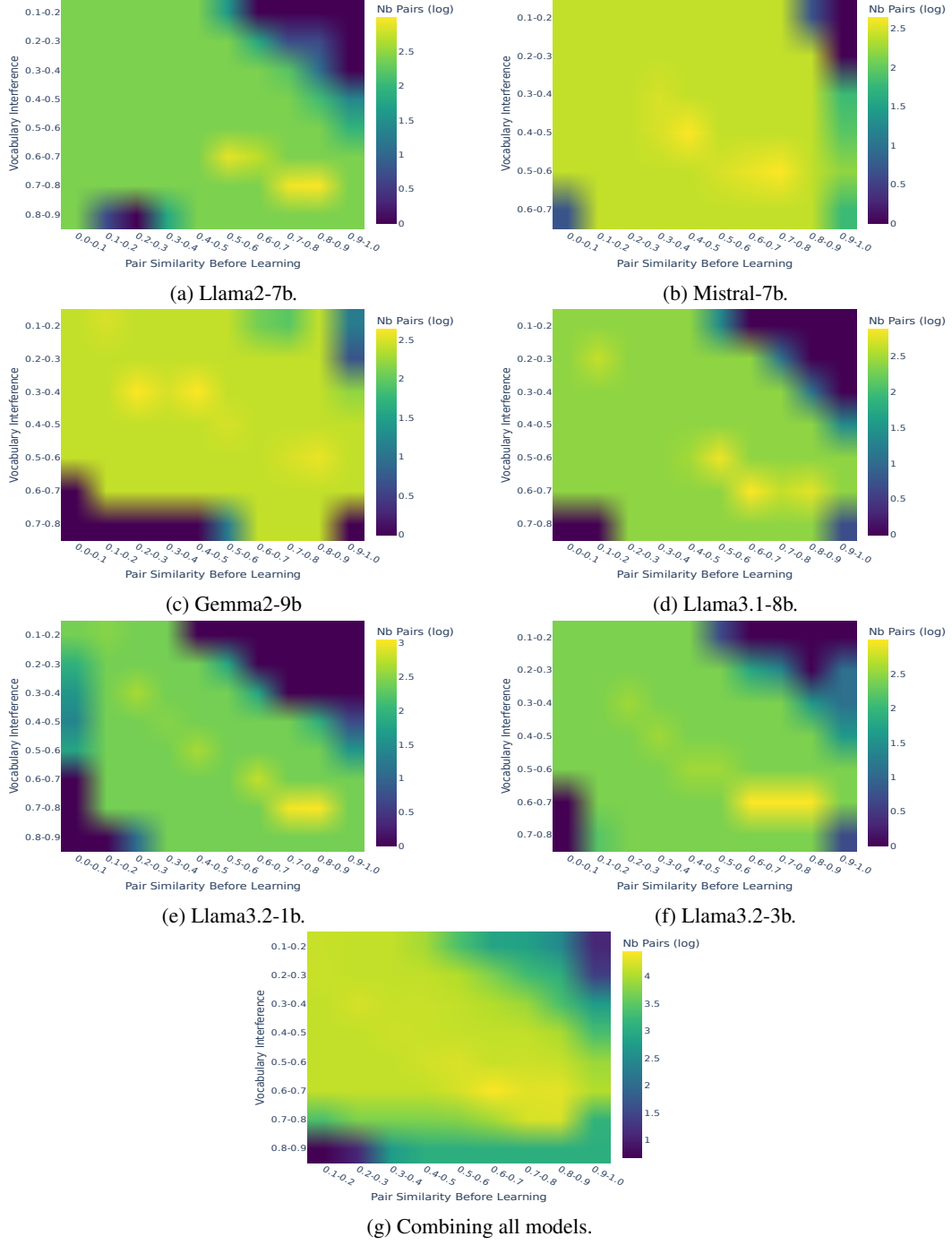


Figure 5: Log-scale heatmap showing the joint distribution of token pairs across pairwise similarity before learning (x-axis) and vocabulary interference (y-axis) after uniformly sampling of 10 items per pairwise similarity  $\times$  vocabulary interference bin. Subplots (a–f) correspond to individual models; subplot (g) aggregates results across all models.

#### A.4 Defining levels of vocabulary interference

Figure 6a shows a kernel density estimate (KDE) of the vocabulary interference scores (median values) computed across all evaluated token pairs. To define the Low, Mid, and High interference categories, we divided the distribution into three quantiles, with the resulting quantile thresholds indicated by the vertical dashed lines. Figure 6b reports the number of token pairs (log scale) used in our analysis, stratified by both vocabulary interference level and pair similarity prior to learning.

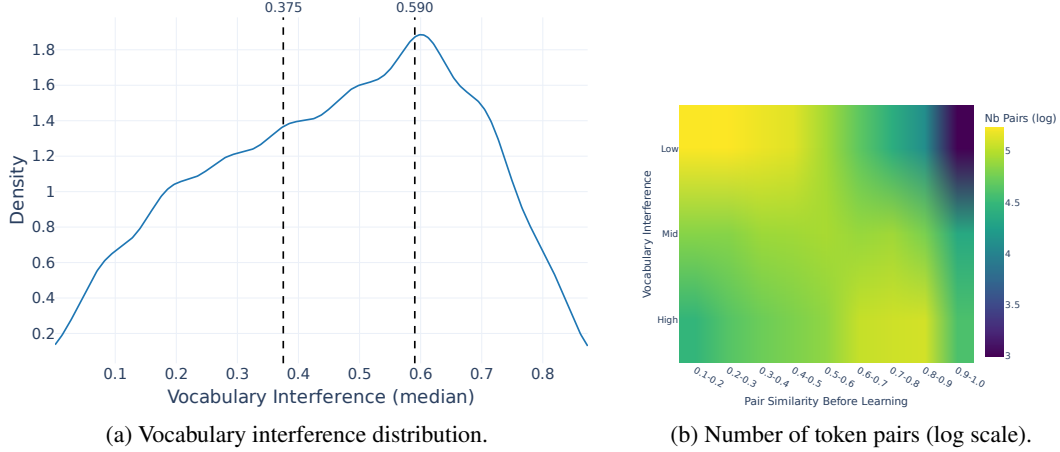


Figure 6: (a) Distribution of vocabulary interference (median) values. Vertical lines show the thresholds used to equally split this distribution into Low, Mid and High similarity levels. (b) Heatmap showing the number of token pairs (log scale) as a function of pairwise similarity before learning (x-axis) and vocabulary interference level (y-axis) after uniformly sampling of 10 items per pairwise similarity  $\times$  vocabulary interference bin.

#### A.5 Modification to Greedy Coordinate Descent (GCG) algorithm

We repurpose the GCG [55] method to minimize a loss defined over the cosine similarity of internal activations. Specifically, we randomly sample a token  $x$  and construct a starting input sequence  $s = [x_1, y_1]$ , where  $x_1 = y_1$ , i.e., the same token is used as starting point in both positions. We then measure their pair similarity,  $S_1^m = \cos(\mathbf{h}_{x_1}^m, \mathbf{h}_{y_1}^m)$ . We fix  $x_1$ , and our goal is to iteratively replace  $y_1$  until the pair similarity converges to the target interval  $[\theta_{\min}^g, \theta_{\max}^g]$ . To find a suitable replacement, we define a loss function for each group to target the midpoint of the interval,  $\mathcal{L}_g = (\frac{\theta_{\max}^g - \theta_{\min}^g}{2} - S_1^m)^2$ . We then compute its gradient with respect to the one-hot encoding of  $y_1$ . This produces a vector indicating how sensitive the loss is to each token in the vocabulary, which we then use to guide the search for a more suitable substitution, without updating the model’s weights.

Next, we identify the top- $k$  ( $k = 256$ ) tokens associated with the steepest decrease in loss (i.e. the most negative gradients). These candidates serve as a rough approximation of the most promising substitutions, obtained via a first-order Taylor expansion. We randomly shuffle this top- $k$  subset and evaluate each candidate sequentially by constructing a modified input sequence and computing the loss for each candidate pair. The candidate yielding the smallest loss (i.e., closest cosine similarity to the target range) is selected as the updated token for  $y_1$  on this iteration. This procedure is repeated for a fixed number of iterations ( $it = 100$ ) or until the similarity score falls within the target interval. If convergence is not achieved within the allotted iterations, the process restarts from a newly sampled initial token pair.

## Appendix B Supplementary analyses of the main paper results

### B.1 Example of token pairs

Group	Pair 1	Pair 2	Pair 3
0.1–0.15	(Liter, CLARE)	(artifactId, gew)	(emat, SOUR)
0.15–0.2	(Ste, UITableView)	(Pers, pmatrix)	(ries, pragma)
0.2–0.25	(it, Autres)	(bt, Autres)	(Bad, tf)
0.25–0.3	(VD, Autres)	(Vertical, ierte)	(coordinate, gesch)
0.3–0.35	(elf, ScrollView)	(Tr, named)	(Else, newcommand)
0.35–0.4	(DER, stackexchange)	(ific, ently)	(von, trightarrow)
0.4–0.45	(uk, ThreadPool)	(vez, ISBN)	(under, rov)
0.45–0.5	(illet, cially)	(icio, atr)	(ptop, Wikimedia)
0.5–0.55	(bootstrap, rach)	(utes, Vorlage)	(iveau, tersuch)
0.55–0.6	(mittel, umbn)	(Series, notify)	(Problem, emptyset)
0.6–0.65	(fte, zott)	(Length, TRUE)	(elve, PDF)
0.65–0.7	(nings, setAttribute)	(isen, issenschaft)	(ouv, schluss)
0.7–0.75	(ru, occup)	(result, utzt)	(aka, rola)
0.75–0.8	(cock, eland)	(hib, heast)	(prepare, Once)
0.8–0.85	(relation, emptyset)	(reen, bmatrix)	(uliar, ienn)
0.85–0.9	(Italie, urre)	(cement, cement)	(onna, onna)
0.9–0.95	(aped, aped)	(loster, loster)	(lict, lict)

Table 2: Examples of token pairs for Llama2-7b.

Group	Pair 1	Pair 2	Pair 3
0.1–0.15	(anity, OptionsMenu)	(attributes, Bitte)	(Commission, OptionsMenu)
0.15–0.2	(Transform, LEncoder)	(orgetown, DataProvider)	(download, SFML)
0.2–0.25	(types, DefaultCloseOperation)	(Defense, Autor)	(Cookies, Magn)
0.25–0.3	(VISION, Nut)	(ansi, Very)	(plants, addAll)
0.3–0.35	(COM, Refer)	(UUFFIX, getResource)	(ModelError, fol)
0.35–0.4	(ifers, findViewById)	(Boundary, Foot)	(Quant, developers)
0.4–0.45	(arena, rak)	(Reuters, inflate)	(replacement, Detail)
0.45–0.5	(container, flu)	(webElementX, Sal)	(yet, multipart)
0.5–0.55	(Mission, Axis)	(down, remark)	(Rails, pictureBox)
0.55–0.6	(tier, messages)	(Mart, bold)	(analytics, Vis)
0.6–0.65	(grunt, pro)	(led, closest)	(matrix, stackpath)
0.65–0.7	(bye, byte)	(zeros, asctime)	(ending, protect)
0.7–0.75	(icated, ensure)	(afka, ref)	(flowers, caption)
0.75–0.8	(classed, classed)	(ERM, NASA)	(icana, mui)
0.8–0.85	(aggio, derive)	(tura, fillna)	(OnClick, endregion)
0.85–0.9	(ylko, echo)	(entario, cite)	(Avoid, inf)
0.9–0.95	(hotel, hotel)	(igrate, cite)	(recio, ulado)

Table 3: Examples of token pairs for Llama3.1-8b.

Group	Pair 1	Pair 2	Pair 3
0.1–0.15	(Flash, ph)	(fake,stantiateViewController)	(Package, rid)
0.15–0.2	(taken,addPreferredGap)	(lambda,stantiateViewController)	(Nintendo, Fre)
0.2–0.25	(Chooser, meye)	(ENDED, queueReusable)	(Phil, NegativeButton)
0.25–0.3	(vehicles, uden)	(ideon, DllImport)	(Inverse, Wel)
0.3–0.35	(pressure, VertexAttrib)	(ForeignKey, gesch)	(those, CLLocation)
0.35–0.4	(Deferred, textAlign)	(sharp, SetBranchAddress)	(DEST, British)
0.4–0.45	(Across, Cas)	(Career, ud)	(Andre, Cod)
0.45–0.5	(oldemort, NumberFormatException)	(Binary, IMITIVE)	(indexes, BitConverter)
0.5–0.55	(represented, toContain)	(Chinese, THIS)	(jk, flatMap)
0.55–0.6	(Bon, Bon)	(Already, dbc)	(iston, Sub)
0.6–0.65	(Exam, let)	(Projectile, iores)	(odie, objectManager)
0.65–0.7	(Americans, illes)	(diff, stderr)	(Consulta, ificantly)
0.7–0.75	(LinkedIn, Prime)	(doctrine, iale)	(Space, vore)
0.75–0.8	(Get, Get)	(bomb, bomb)	(stripe, rightarrow)
0.8–0.85	(identifier, identifier)	(roller, roller)	(dimension, Intialized)
0.85–0.9	(Width, Width)	(Kitchen, Kitchen)	(landers, landers)
0.9–0.95	(Iterator, Iterator)	(balanced, balanced)	(pricing, pricing)

Table 4: Examples of token pairs for Llama3.2-1b.

Group	Pair 1	Pair 2	Pair 3
0.1–0.15	(Great, getKey)	(Warnings, toEqual)	(Direct, Ser)
0.15–0.2	(Water, Ref)	(Pop, CREATE)	(sole, NET)
0.2–0.25	(children, ischen)	(hentic, redirect)	(TEMP, operatorname)
0.25–0.3	(username, por)	(Permission, LECT)	(submit, riev)
0.3–0.35	(JSON, optim)	(objects, junit)	(cat, contr)
0.35–0.4	(good, resolve)	(sam, partition)	(glas, forEach)
0.4–0.45	(Bank, Thank)	(append, stri)	(Interval, Mic)
0.45–0.5	(One, One)	(CEPT, Accept)	(ervices, Reference)
0.5–0.55	(Must, Must)	(Temp, Temp)	(foo, hbar)
0.55–0.6	(testing, testing)	(Input, Selector)	(PATH, THE)
0.6–0.65	(size, size)	(friend, mary)	(LowerCase, pathy)
0.65–0.7	(ebook, ebook)	(itzer, sender)	(LIB, LIB)
0.7–0.75	(century, century)	(ted, ted)	(JSON, ensuremath)
0.75–0.8	(Azure, Azure)	(aws, aws)	(ton, ton)
0.8–0.85	(getInt, getInt)	(uff le, ible)	(jem, jem)
0.85–0.9	(backup, backup)	(strlen, strlen)	(Width, Width)
0.9–0.95	(NonNull, NonNull)	(jed, jed)	(urd, urd)

Table 5: Examples of token pairs for Mistral-7b.

Group	Pair 1	Pair 2	Pair 3
0.1–0.15	(Wednesday, Ngh)	(right, NSMutable)	(Pear, FILES)
0.15–0.2	(particle, Nej)	(easy, unc)	(Trader, multip)
0.2–0.25	(berry, Incre)	(Drawer, requ)	(OPTIONS, Fil)
0.25–0.3	(Life, bef)	(Reward, coc)	(Pages, Jer)
0.3–0.35	(your, enc)	(Thickness, atab)	(widgets, ilerek)
0.35–0.4	(borrow, empre)	(Restart, hatt)	(Crypto, orm)
0.4–0.45	(bul, dney)	(Creates, olumn)	(bout, OrNull)
0.45–0.5	(Hola, vert)	(ops, olare)	(Companies, SuppressWarnings)
0.5–0.55	(Fall, comm)	(Transient, bold)	(affected, big)
0.55–0.6	(Informe, sys)	(high, big)	(anna, badge)
0.6–0.65	(thought, Tags)	(Experiment, operator)	(sure, isset)
0.65–0.7	(yellow, center)	(empty, tiny)	(creen, rather)
0.7–0.75	(answers, prompt)	(Seats, Seats)	(Ryan, Ryan)
0.75–0.8	(sets, sets)	(ordinal, ordinal)	(conte, conte)
0.8–0.85	(telegram, telegram)	(Israeli, Israeli)	(fontsize, fontsize)
0.85–0.9	(country, country)	(pokemon, pokemon)	(gmail, gmail)
0.9–0.95	(PlainText, PlainText)	(bbc, bbc)	(jquery, jquery)

Table 6: Examples of token pairs for Llama3.2-3b.

Group	Pair 1	Pair 2	Pair 3
0.1–0.15	(logged, Zapraszamy)	(pharmacy, SuspendLayout)	(Closing, CODES)
0.15–0.2	(al, population)	(Readers, charging)	(brink, setObjectName)
0.2–0.25	(Prev, Findings)	(Knowledge, Whip)	(Detalle, WriteTagHelper)
0.25–0.3	(Colin, Lauren)	(favor, Aunt)	(Wikimedia, mechanical)
0.3–0.35	(few, trust)	(networking, StructEnd)	(luxe, Williams)
0.35–0.4	(Attribute, Angels)	(Hotline, Bought)	(ScrollView, archiviato)
0.4–0.45	(Mila, conductor)	(cian, river)	(zhen, Really)
0.45–0.5	(Holly, Nicole)	(Runners, Anybody)	(politics, Shown)
0.5–0.55	(Rejection, Accreditation)	(association, assembler)	(voiture, Document)
0.55–0.6	(developing, specifically)	(nvidia, codegen)	(pressing, scribes)
0.6–0.65	(assistance, expliquer)	(METHOD, CARD)	(MouseMove, cooperation)
0.65–0.7	(covariance, collection)	(markup, Upgrade)	(monger, metrist)
0.7–0.75	(COMPLEX, TRUST)	(ExecuteReader, MemoryWarning)	(dima, dima)
0.75–0.8	(listBox, errorMessage)	(Commit, Commit)	(Flesh, Flesh)
0.8–0.85	(ModelAndView, Element)	(componentWill, componentWill)	(navigateTo, navigateTo)
0.85–0.9	(tapete, felpa)	(ByteString, Interface)	(mapreduce, mapreduce)
0.9–0.95	(getSelection, getSelection)	(Salmo, Salmo)	(ido, ido)

Table 7: Examples of token pairs for Gemma2-9b.

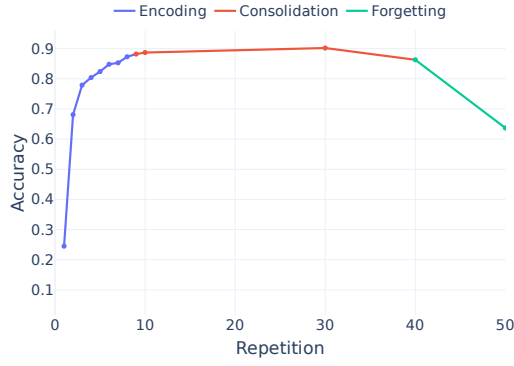
## B.2 Accuracy dynamics per model

In the main text (Figure 2a), we visualize how model accuracy evolves across different training stages by segmenting the process into three distinct phases: Encoding, Consolidation, and Forgetting. Since models may undergo a different number of repetitions in each phase, we normalized the  $x$ -axis by mapping each phase to a fixed interval. This temporal alignment enables meaningful comparison of performance trajectories across models on a shared timeline. In Table 8, we provide details on when the learning phase transitions occur, and show the performance of each model at those transitions. Figure 7 shows the accuracy dynamics across repetitions for all models, up to 50 repetitions. We can observe that each model has a slight different learning dynamic.

Table 8: Model performance (i.e., accuracy on the associative task) across learning phases. For each model, we report the accuracy and repetition: at the end of the Encoding  $\rightarrow$  Consolidation phase, at the maximum accuracy achieved during Consolidation, and at the end of Consolidation  $\rightarrow$  Forgetting phase when applicable.

Model	Encoding $\rightarrow$ Consolidation	Max. Accuracy	Consolidation $\rightarrow$ Forgetting
Gemma2-9b	0.98 ( $r = 3$ )	1.0 ( $r = 30$ )	-
Llama3.2-1b	0.96 ( $r = 5$ )	1.0 ( $r = 150$ )	-
Llama3.2-3b	0.97 ( $r = 6$ )	1.0 ( $r = 150$ )	-
Llama3.1-8b	0.97 ( $r = 4$ )	1.0 ( $r = 100$ )	-
Llama2-7b	0.87 ( $r = 8$ )	0.9 ( $r = 30$ )	0.86 ( $r = 40$ )
Mistral-7b	0.96 ( $r = 8$ )	1.0 ( $r = 600$ )	0.83 ( $r = 3k$ )

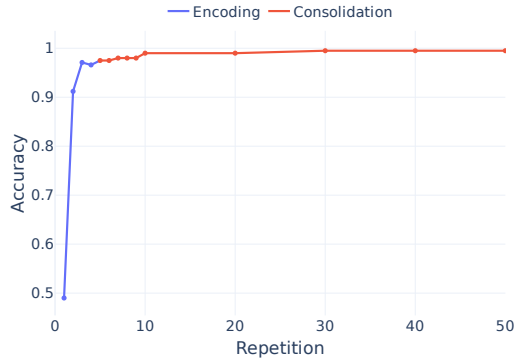




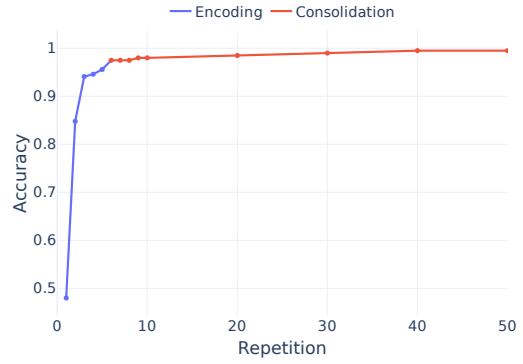
(a) Llama2-7b.



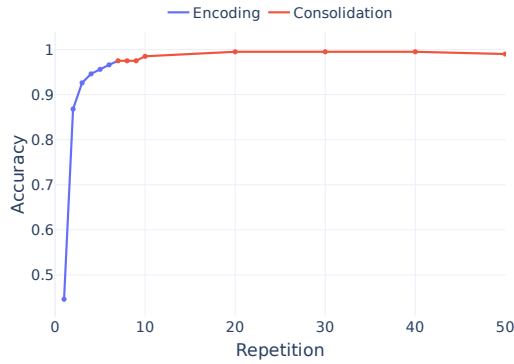
(b) Mistral-7.



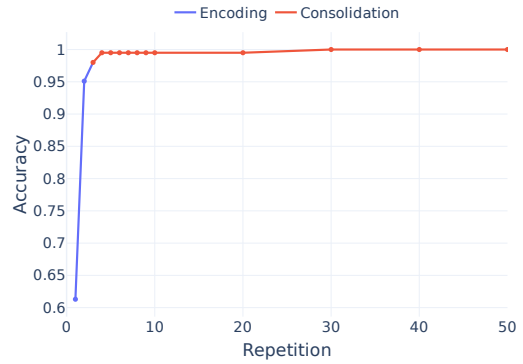
(c) Llama3.1-8b.



(d) Llama3.2-1b.



(e) Llama3.2-3b.



(f) Gemma2-9b.

Figure 7: Accuracy over repetitions for all model, shown up to 50 repetitions.

### B.3 Representation dynamics per model

In the main text (Figure 2b), we present normalized trajectories of representational change across learning phases, allowing comparison across models. Here, we provide the corresponding per-model plots (Figure 8), showing representational change across repetitions. Across models, we observe a consistent non-monotonic trend during the Consolidation (straight line) phase.



Figure 8: Representational changes across repetitions and their corresponding learning phase (one plot per model). To reduce an overly dense visualization, we display a subset of repetitions: for models with a forgetting phase, 2 repetitions per phase were selected; for models without a forgetting phase, 3 repetitions per phase were included. Across all models, we observe a non-monotonic trend aligned with NMPH during the consolidation phase.

## B.4 Potential factors in forgetting phase

In the main text (Section 4.1), we observed that two models—Llama2-7b and Mistral-7b—showed a forgetting phase, characterized by a drop in accuracy greater than 3% relative to the average of the two preceding repetitions. This behavior indicates the start of performance degradation. We speculate that the delayed forgetting observed in Mistral-7b may be influenced by its use of a sliding window attention (SWA) mechanism.

We performed an initial analysis of a possible—though speculative—factor that may have influenced the forgetting phase observed in the Llama2-7b model. Figure 9 shows the distribution of vocabulary interference, where vertical lines show the average pair similarity after learning, per group. The subfigures show the distribution for the last repetition of the Consolidation ( $r = 30$ ) and the first repetition of the Forgetting phases ( $r = 40$ ), respectively. Notably, during Forgetting, token pairs shift toward the peak of the interference distribution. This suggests that Forgetting occurs when there is increased competition from similar vocabulary items, which could be impairing the model’s ability to maintain accurate associations. This interpretation remains speculative, and future work could further investigate the causes of forgetting and their relationship to interference.

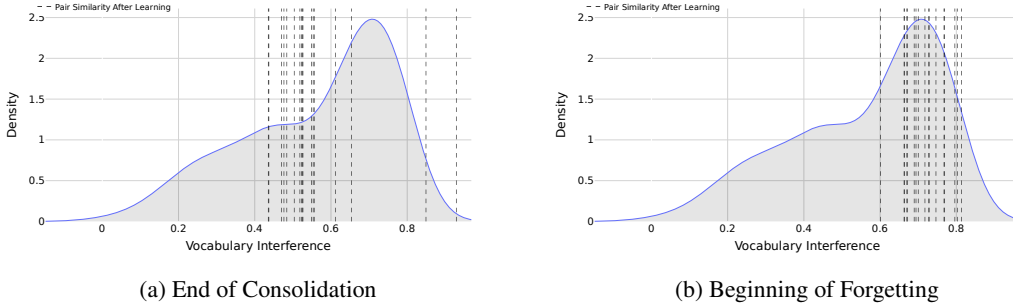


Figure 9: Vocabulary interference distribution for Llama2-7b at (a) the end of the Consolidation phase ( $r = 30$ ), and (b) the start of the Forgetting phase ( $r = 40$ ). Vertical dashed lines indicate the average pair similarity after learning for each group. During Forgetting, a noticeable shift in pair similarity toward the peak of the interference distribution suggests increased competition, potentially contributing to the observed decline in performance.

## B.5 Supplementary analyses of representational dynamics

Figure 10 shows the trajectory of representational change across learning phases separately for low, moderate, and high similarity groups. The mid-similarity group includes only those pairs that exhibited significant differentiation in the t-test analysis from Section 4.2. Low- and high-similarity categories were defined by aggregating the remaining pairs based on their similarity scores. The results reveal distinct dynamics across similarity regimes, although the overall shape of the changes remains consistent across similarity groups. Low-similarity pairs exhibit a sharp increase in representational similarity during the initial repetitions of the Encoding phase, followed by a gradual decline throughout Consolidation. In contrast, mid-similarity pairs show a more modest increase during Encoding but undergo a significant decrease during Consolidation, ultimately exhibiting strong differentiation. High-similarity pairs remain relatively stable, with only a slight increase during Encoding and a minor reduction during Consolidation. These trends are broadly consistent across models.

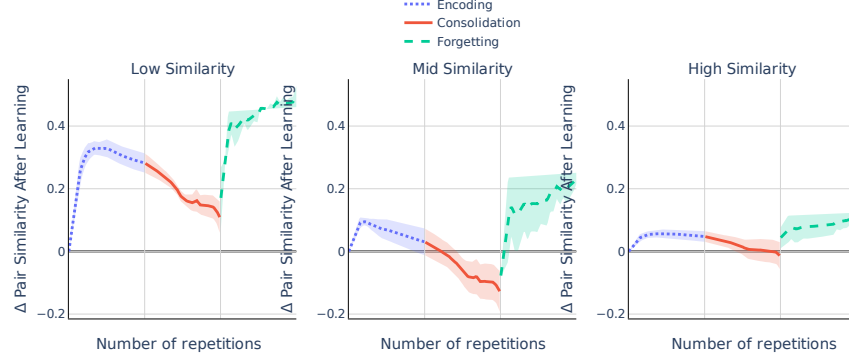


Figure 10: Representational change across learning phases (Encoding, Consolidation, Forgetting) for different pairwise similarity categories. Mid-similarity pairs were selected based on the groups that showed significant differentiation in our t-test analysis (Figure 2b). All groups with lower similarity scores were aggregated into the low-similarity category, and those with higher scores into the high-similarity category. Data is averaged across models. Shaded areas represent the standard error across models.

## B.6 Analysis for extended set

We extended the main analysis to search for 100 token pairs per similarity group, for both Llama2-7b and Llama3.2-1b. The results reveal consistent patterns with those shown in Figure 2 of the main paper.

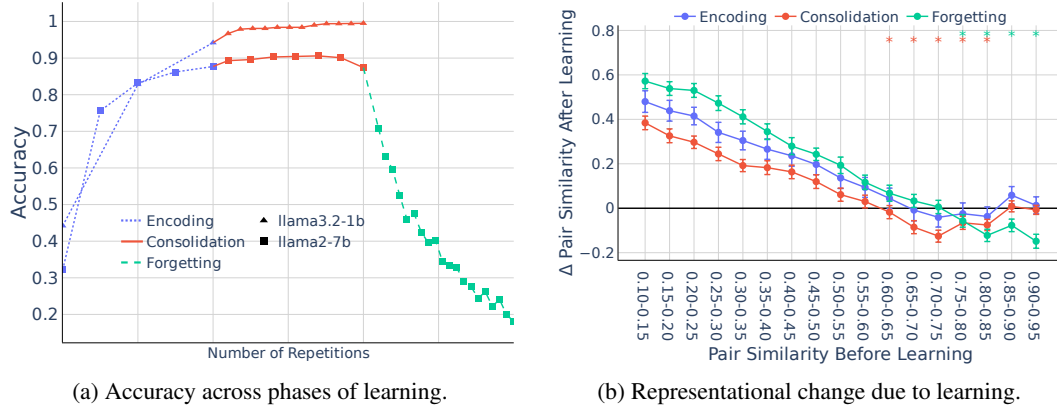


Figure 11: Accuracy and representational changes during learning for an extended stimulus set comprising 100 optimized token pairs in each of the 17 similarity groups. (a) Models show three phases of learning: encoding, where accuracy steeply increases; consolidation, where accuracy stabilizes; and forgetting, where accuracy declines. The x-axis for each model is scaled by the length of its learning phase. (b) The U-shaped differentiation pattern, characteristic of the Non-Monotonic Plasticity Hypothesis, is observed only during consolidation (red). Asterisks (\*) indicate groups that remain significant after Benjamini-Yekutieli correction for multiple comparisons across similarity groups and phases ( $q < 0.05$ ).

## Appendix C Analysis for WordNet token pairs

Our study intentionally selected token pairs selected for their pair similarity before learning, regardless of semantic meaning. This design mirrors the use of synthetic stimuli in [46], which intentionally avoids meaningful real-world inputs and emphasizes the importance of sampling across the entire similarity spectrum—especially the mid-similarity range—to effectively test NMPH. Because real-world tokens are unevenly distributed across this space, achieving precise control is otherwise difficult. Accordingly, our primary aim in this work is not to study meaning, but to examine the structural dynamics of representational change in response to learning.

That said, in this section we briefly assess how representation dynamics evolve under more naturalistic conditions. In our main analyses, we already filtered out tokens containing numbers, punctuation, or special characters. Here, we further restricted token pairs to single-token WordNet words, which reduced the usable vocabulary to roughly  $\approx 2.4\text{k}$  tokens out of  $\approx 28\text{k}$  for Llama2-7b and  $\approx 4.8\text{k}$  out of  $\approx 10\text{k}$  for Llama3.2-1b.

We first examined how pairwise similarity and vocabulary interference were distributed within this constrained space, anticipating that the reduced vocabulary might bias pairs toward narrower interference ranges. Indeed, sampled real-world token pairs show higher vocabulary interference than our synthetic token pairs (Figure 12), suggesting that they face stronger competition during prediction. Our results (Figure 13) confirmed this: similarity after learning decreased monotonically with respect to the similarity before learning, supporting the view that highly similar pairs are modulated by vocabulary interference. Together, these findings suggest that global interference is a key factor modulating representational dynamics in naturalistic learning settings, and that NMPH emerges under specific conditions of global interference.

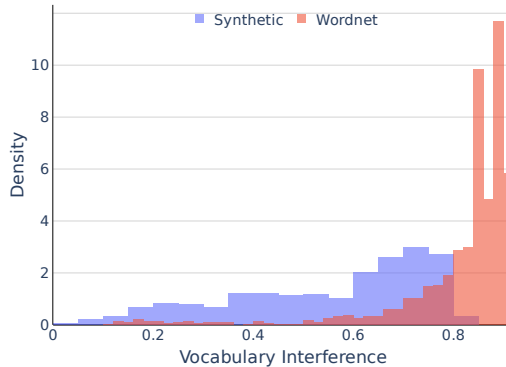
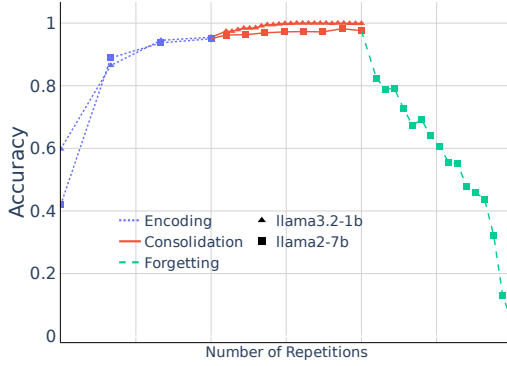
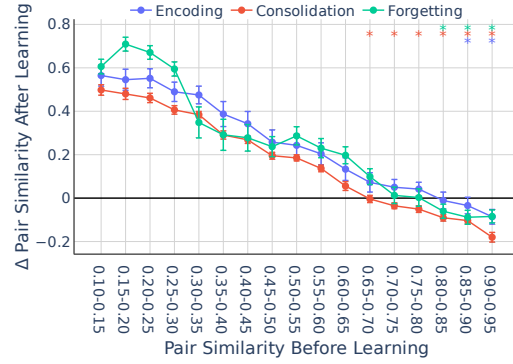


Figure 12: Distribution of vocabulary interference for previously-optimized synthetic token pairs versus WordNet token pairs. Original pairs (blue) span the full similarity spectrum, enabling controlled sampling across ranges, while WordNet pairs (red) cluster at high vocabulary interference values. This skew highlights the difficulty of achieving balanced coverage with real-world tokens and motivates the use of optimized and more synthetic stimuli to test NMPH.



(a) Accuracy across phases of learning.



(b) Representational change due to learning.

Figure 13: Accuracy and representational changes during learning with a set of WordNet token pairs. (a) Across models, learning unfolds in three phases: Encoding, marked by a steep rise in accuracy; Consolidation, where accuracy stabilizes; and Forgetting, where accuracy declines. The x-axis for each model is scaled to the length of its learning phase. (b) In contrast to earlier results, the characteristic U-shaped differentiation pattern is diminished, giving way to a monotonically decreasing trend, consistent with the higher vocabulary interference observed among real-word token pairs. Asterisks (\*) denote similarity groups that remain significant after Benjamini–Yekutieli correction for multiple comparisons across groups and phases ( $q < 0.05$ ).

### C.1 WordNet token pair examples

Similarity Range	Pair 1	Pair 2	Pair 3
0.1–0.15	(mix, loaded)	(defined, standard)	(pub, any)
0.15–0.2	(identifier, astern)	(layout, eclipse)	(defined, slash)
0.2–0.25	(online, pop)	(suite, abb)	(pub, format)
0.25–0.3	(absolute, pus)	(round, pa)	(annotation, hum)
0.3–0.35	(series, math)	(black, roc)	(gas, bat)
0.35–0.4	(spec, cock)	(information, leg)	(argument, lear)
0.4–0.45	(contra, architecture)	(dictionary, ike)	(rooms, ho)
0.45–0.5	(gen, nil)	(factory, acre)	(shadow, nih)
0.5–0.55	(gen, raise)	(time, eb)	(zero, iga)
0.55–0.6	(dale, person)	(dawn, esp)	(irs, ante)
0.6–0.65	(any, essen)	(final, mission)	(gi, dim)
0.65–0.7	(cap, bind)	(mus, skim)	(dd, safe)
0.7–0.75	(bye, anas)	(izar, through)	(lined, click)
0.75–0.8	(replace, stock)	(unction, week)	(execution, frame)
0.8–0.85	(geometry, list)	(locale, embed)	(partition, brand)
0.85–0.9	(opacity, fragment)	(render, inflate)	(analysis, section)
0.9–0.95	(gable, board)	(volution, ship)	(slider, simple)

Table 9: Wordnet token pairs examples for llama2-7b.

Similarity Range	Pair 1	Pair 2	Pair 3
0.10–0.15	(tour, rather)	(elect, subscribe)	(speaker, hear)
0.15–0.20	(inherit, soon)	(phone, six)	(internal, town)
0.20–0.25	(access, version)	(roman, doll)	(artist, then)
0.25–0.30	(import, traffic)	(flat, sin)	(license, raj)
0.30–0.35	(creator, solution)	(department, ne)	(package, ghost)
0.35–0.40	(use, company)	(declare, dead)	(linux, gu)
0.40–0.45	(sign, oracle)	(google, cro)	(district, bone)
0.45–0.50	(code, rabbit)	(sign, testing)	(public, edd)
0.50–0.55	(code, extended)	(code, radius)	(code, folder)
0.55–0.60	(far, match)	(sea, ledger)	(type, memory)
0.60–0.65	(wide, resize)	(sea, timing)	(dot, sector)
0.65–0.70	(express, window)	(sky, connection)	(mind, league)
0.70–0.75	(identifier, technical)	(mind, oracle)	(earth, corner)
0.75–0.80	(dream, burst)	(pixel, circle)	(earth, setter)
0.80–0.85	(beer, burst)	(moon, window)	(shirt, issue)
0.85–0.90	(ticker, check)	(poser, former)	(ticker, heartbeat)
0.90–0.95	(badge, piece)	(widget, pillar)	(spender, heading)

Table 10: WordNet token pair examples for Llama3.2-1b.

## Appendix D Analysis of other layers of the models

### D.1 Additional improvements to token pair search algorithm for obtaining earlier layer representations

While the procedure described in Section A.5 identified suitable pairs across a range of similarities when we looked at hidden representations from the last layer of each LLM, some convergence issues arose when we explored representations in earlier layers. In particular, selecting tokens with the most negative gradients did not consistently decrease the loss over repeated iterations. We reasoned that this may be equivalent to taking step sizes that are too large in the gradient descent. To remedy this, we modified our procedure to add line search backtracking to impose a bound on the gradient, only selecting candidate tokens with gradients between  $[0, -bound]$  [23]. If a given iteration does not decrease the loss a sufficient amount (under the Armijo condition,  $\alpha = 0.3$ ), the step is rejected. The gradient bound is then brought closer to 0 by a factor of  $\beta = 0.2$ , until it reaches the maximum value of  $1e - 8$ . The best candidate pair  $[x_1, y_1]$  based on the smallest loss is kept across iterations.

If a given starting token  $x_1$  does not converge after 100 iterations, we add the best candidate pair to the similarity group that it falls into (if the group is not already full).

### D.2 Representational change using stimuli optimized for earlier layers

**Optimization setup.** We searched for stimulus pairs using representations from earlier layers in 3 models: llama2-7b, mistral-7b, and gemma2-9b. We chose to evaluate 2 layers each from early, middle and late positions in the model, for a total of 6 layers. Early layers were always layers 1 and 2. The middle layers began at half the number of total layers (which varied between models), and the one after that. The late layers corresponded to layer indices  $-3$  and  $-2$ , directly preceding the last layer that we analyzed in the main text.

We were able to find the full set of stimulus pairs (12 pairs per group) in the similarity interval  $[0.1 - 0.8)$ , but were less successful for the high similarity groups. The number of total stimulus pairs per group is given in Table 11.

Table 11: Number of stimuli found in each similarity group for each learning phase and earlier layer, summed across the 3 models.

Phase	Layer	Similarity group					
		0.1-0.15	...	0.75-0.8	0.8-0.85	0.85-0.9	0.9-0.95
Encoding	early	528		528	398	197	95
	mid	540		540	400	202	74
	late	480		480	370	167	55
Consolidation	early	1440		1440	1230	693	395
	mid	1368		1368	1168	656	390
	late	1428		1428	1198	675	409
Forgetting	early	696		696	616	292	50
	mid	756		756	676	324	76
	late	756		756	676	340	76

As expected, the accuracy on the task remained about the same when using stimuli optimized for similarity in earlier layers (Figure 14).

**Earlier layer results.** We then extracted hidden representations in two complementary ways.

First, we analyzed token pairs that were optimized for token pair similarity in earlier layers (Figure 15a). This allowed us to assess representational changes at intermediate depths of the model, relative to the pair similarity before learning for which the pairs were optimized. In these analyses, intermediate and late layers exhibited a largely monotonic decrease in similarity, with pronounced differentiation for pairs with high pair similarity before learning ( $>0.7$ ). Differentiation effects



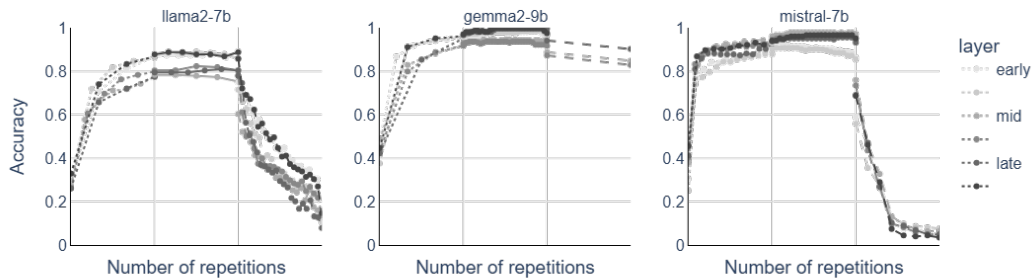
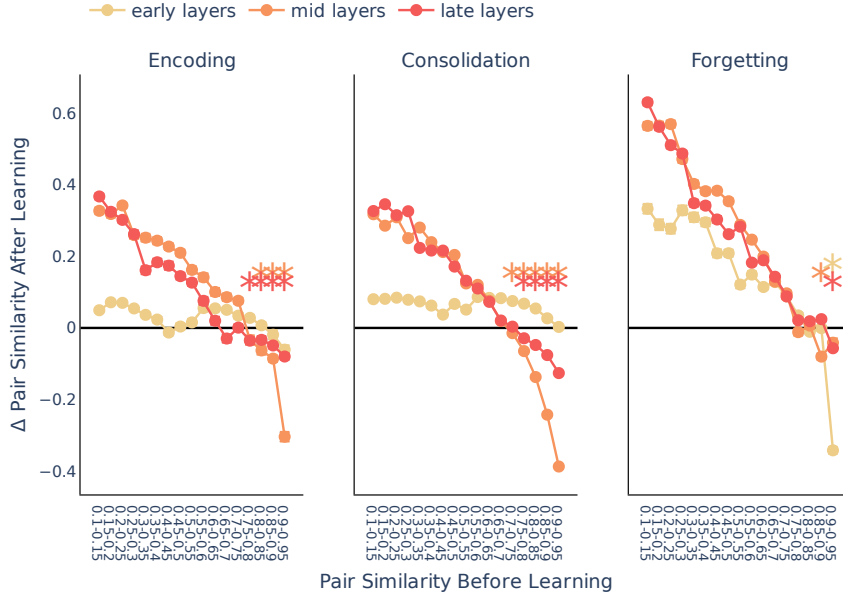


Figure 14: Stimuli optimized for representational similarity in other layers maintains similar accuracy on the associative learning task.

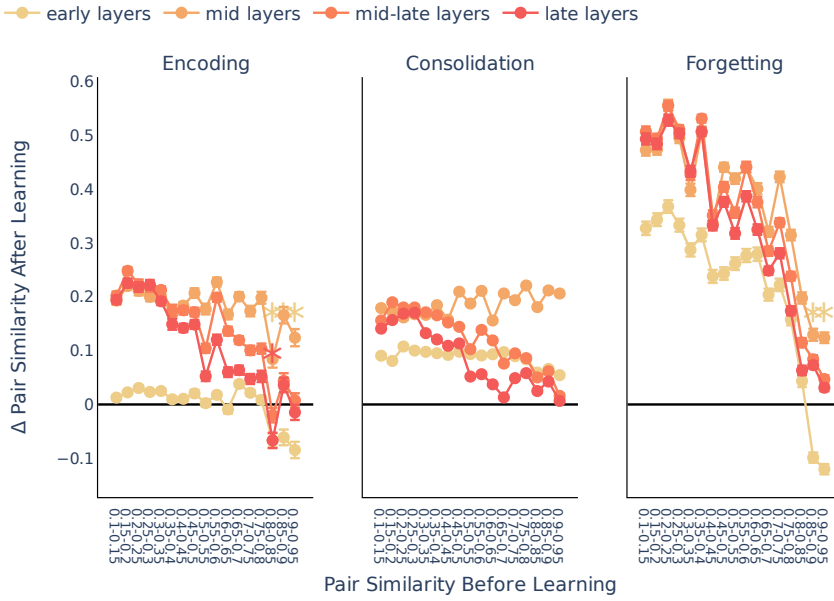
were stronger in mid layers than in late layers, whereas the earliest layers (1 and 2) behaved more erratically and did not display a consistent trend.

Second, we evaluated the same set of token pairs that had been optimized for the last layer (as in Figure 2b) but measured their representational changes across earlier layers (Figure 15b). This analysis was designed to track how the non-monotonic pattern observed at the output layer emerges progressively across the model hierarchy. During the consolidation phase, early to mid layers showed relatively flat or mildly monotonic decreasing trends, with similarity values remaining above zero and thus reflecting representational integration. Mid-late layers began to show a clearer monotonic decrease in similarity. In the final layers, the emergence of a non-monotonic, U-shaped pattern was visible, although the minimum of the curve did not correspond to statistically significant differentiation. Taken together, these findings suggest that representations initially integrate across similarity levels and gradually develop the U-shaped structure as they propagate through the model depth.

Finally, we examined the role of vocabulary interference as a potential driver of this effect. We observed (Figure 16) a general increase in global interference with layer depth, such that deeper layers face stronger competition among possible token predictions. This increasing interference provides a plausible mechanism for the stronger differentiation observed in later layers, supporting the interpretation that global interference modulates the representational change pattern.



(a) Token pairs optimized for earlier layers.



(b) Token pairs optimized for the last layer.

Figure 15: (a) Representational change across model layers for token pairs optimized at early layers. Early layers (1-2) exhibit irregular and non-systematic changes in similarity, suggesting unstable representations. Intermediate and late layers show a more consistent monotonic decrease—particularly for highly similar pairs ( $>0.7$  pair similarity before learning)—with intermediate layers showing stronger differentiation than late layers. (b) Representational change across model layers for token pairs optimized at the last hidden layer. During the consolidation phase, early to mid layers exhibit relatively flat or mildly monotonic decreases in similarity, reflecting representational integration. In contrast, mid-to-late layers begin to show clearer monotonic decreases, and the final layers display the emergence of a U-shaped, non-monotonic pattern. Although the minimum of the curve is not statistically significant, these results suggest that representations integrate at earlier stages and progressively develop non-monotonic structure with increasing model depth.

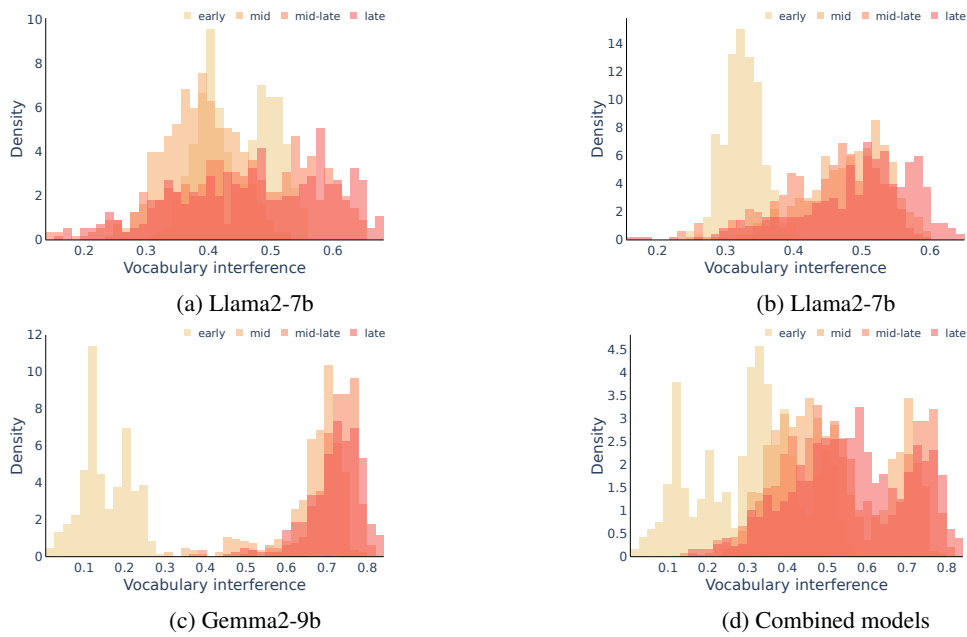


Figure 16: Distribution of vocabulary interference across layers. Global interference increases with layer depth, indicating that deeper layers face stronger competition among possible token predictions. This trend provides a potential mechanism for the stronger differentiation observed in later layers, supporting the interpretation that global interference modulates representational change.