Concept Training for Human-Aligned Language Models

Anonymous ACL submission

Abstract

The next-token prediction (NTP) objective trains language models to predict a single continuation token at each step. In natural language, however, a prefix can be continued in many valid ways, and even similar meanings may differ in surface form. For example, the sentence "this website is safe to browse" could plausibly continue with words such as browse, search, visit, surf, or navigate. While standard NTP training treats these alternatives as mutually exclusive targets, we explore a framework that instead predicts concepts, approximated as sets of semantically related tokens. We show that models trained with concept supervision exhibit stronger alignment with human semantic similarity judgments on multiple lexical benchmarks. These gains are accompanied by lower perplexity on semantically meaningful words (definition in Section 3.1), and a modest increase in global token-level perplexity, reflecting a tradeoff between standard NTP optimization and concept-level supervision. Our results suggest that concept-level objectives can improve semantic alignment while maintaining competitive language modeling performance.[1]

1 Introduction

Large language models (LLMs) generate coherent text, answer questions, and perform increasingly complex reasoning tasks. Yet, the training objective underlying most state-of-the-art LLMs remains strikingly limited: predicting the next token in a sequence. Although the next token prediction (NTP) objective has been successful in practice, it restricts LLMs to learning surface-level token patterns. Specifically, NTP treats different surface realizations as distinct outcomes, even when they convey the same or very similar meaning.

In contrast, humans reason and communicate with concepts that may be realized by variations in phrasing and context. Synonyms such as "dad" and "father" refer to the same conceptual entity, and words that are not exact synonyms can realize the same concept in certain contexts. For instance, "dog" and "pet" can often be interchangeable. As a result, NTP may penalize one valid completion in favor of another, which diverges from the way humans semantically process concepts (Holtzman et al., 2021; Shani et al., 2026).

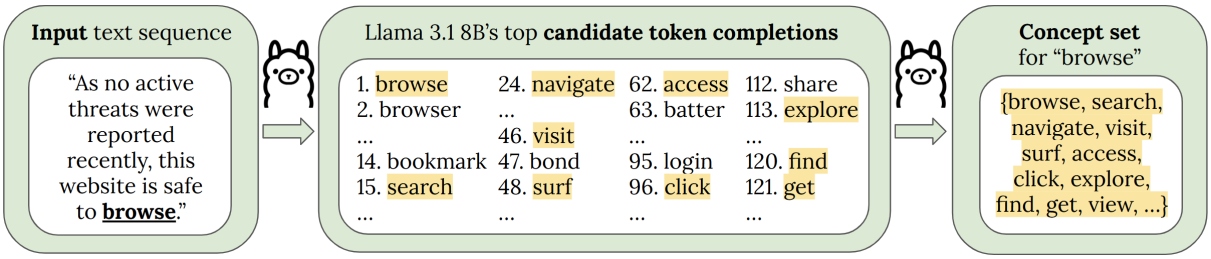
Here, we modify the training objective to **predict concepts rather than surface-level tokens**, where we approximate concepts as sets of contextual synonyms. We introduce a self-contained training pipeline that leverages the model's own knowledge to make its representations more concept-aware.

Our concept-trained models show **stronger alignment with human semantic intuition**. Moreover, despite not being trained to predict exact tokens, concept models achieve lower perplexity and higher accuracy on content words (definition in Section 3.1) than both pretrained and NTP-trained baselines. At the same time, global token-level perplexity increases, suggesting that concept-level predictions shift models away from mere NTP.

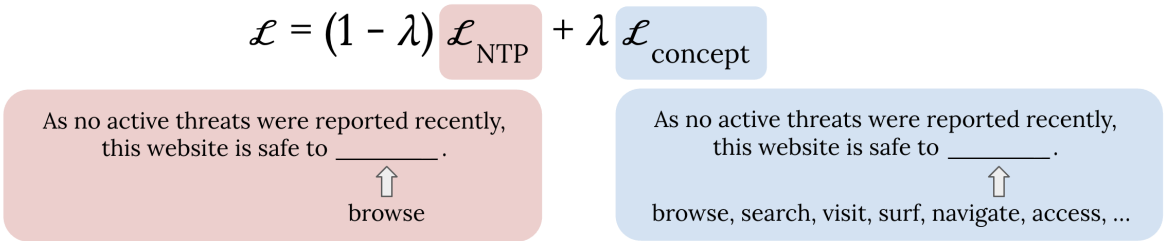
2 Related Work

NTP's tendency to reward only exact token matches has been shown to penalize correct concepts, distort token rankings, and produce undesirable behaviors (Holtzman et al., 2021; Shani et al., 2023; Iyer et al., 2026; Liu et al., 2026). Bender and Koller (2020) argue that this surface-level focus limits a model's ability to learn true semantics, emphasizing the distinction between meaning and form. Empirical work supports this, showing that NTP and teacher-forcing can fail on tasks requiring entailment, planning, or multiple valid continuations (Bachmann and Nagarajan, 2024; Merrill et al., 2024). Together, these suggest that fluent generation does not necessarily imply semantic understanding.

[1] Our code is available at https://anonymous.4open.science/r/concepts-FA2E.



(a) Our process for concept dataset generation, illustrated on one word for this sequence. We pass each sequence into Llama 3.1 8B, and for each noun, verb, or adjective token that comprises an entire word, we extract the top 200 NTP completions. Then, we prompt Llama 3.1 8B Instruct to select the best contextual synonyms for the content words (definition in Section 3.1) in each sequence. These synonyms form the concept set for the original word in the sequence.



(b) Our loss function incorporating both the NTP objective and the concept objective, with the tradeoff tuned via a concept weight hyperparameter λ. Setting $\lambda = 0$ is pure NTP training, and increasing λ makes the models more concept-oriented.

Figure 1: **Concept Training Pipeline**. (a) depicts the data processing phase for enriching it with concept signal, and (b) shows the loss function modification.

Recent work has explored modifying *pretraining* to address these limitations. A survey by Wyatt et al. (2025) categorizes these approaches into: multi-token prediction (Liu et al., 2024; Ahn et al., 2025), plan-then-generate frameworks (Mahajan et al., 2026; Wei et al., 2025), latent reasoning approaches (Barrault et al., 2024; Tack et al., 2026), continuous generation (Li et al., 2022; Shabalin et al., 2025), and non-transformer architectures (Gu and Dao, 2024; Huang et al., 2026).

Others focused on mitigating NTP's limitations during *post-training*. Alignment techniques such as RLHF and DPO (Ouyang et al., 2022; Wu et al., 2023; Rafailov et al., 2023), instruction tuning, contrastive learning, and multitask finetuning introduce semantically informed signals during finetuning. For example, semantic augmentations during instruction tuning improve generalization (Yan et al., 2024), and injecting lexical information improves word sense disambiguation (Levine et al., 2020).

The most closely related approaches incorporate concept-level information directly into the training signal. Chen et al. (2025) propose Concept-Aware Fine-Tuning, which augments NTP with auxiliary heads predicting future tokens. Similarly, Iyer et al. (2026) introduce next-synonym and hypernym prediction methods that enrich training data with semantically related alternatives. Although these methods improve perplexity and downstream performance, they rely on external resources to define concept sets. In contrast, we recover concept structure directly from the model, requiring no external supervision.

3 Training Methods

We now describe both the process of enriching standard data with concept signal (Section 3.1; Figure 1a), and modifying standard NTP objective function to be concept-aware (Section 3.2; Figure 1b).

3.1 Concept Dataset Creation

We used the standard training datasets: 1) AllenAI's C4 (Raffel et al., 2020), and 2) OpenWebText (Gokaslan and Cohen, 2019), an open-source WebText replication. From each corpus, we downsample 2,000 English text sequences, keeping only sequences with 50 < characters < 2,000.

Figure 1a depicts the data generation process. For each sequence, we identify tokens that correspond to complete words (rather than subwords) and are nouns, verbs, or adjectives, ensuring that meaningful semantic alternatives exist. We first tokenize each sequence using the Llama 3.1 8B tokenizer and retain only tokens that form com-

plete words, determined by verifying that both the current token and the following token begin with a space. We then apply spaCy part-of-speech tagging to identify nouns, verbs, and adjectives. To ensure consistency between tokenizations, we verify that these tokens align between spaCy's tokenizer and the Llama 3.1 8B tokenizer. Using this procedure, complete-word single-token nouns, verbs, and adjectives constitute on average 28% of tokens in each sequence. We name these tokens *content words*.

Then, we use Llama 3.1 8B (Grattafiori et al., 2024) to generate the top 200 most likely NTPs for each of these word positions. These candidates represent potential surface forms of the same underlying meaning. Next, we filter this pool to identify contextually interchangeable tokens by prompting Llama 3.1 8B Instruct to select which candidates could replace the original token without changing the meaning of the sentence (Appendix A). The resulting subset forms a contextual synonym set, which we treat as a discrete approximation of the concept associated with that token in its context.

3.2 Concept Training Objective

We introduce a *concept-level loss* that rewards semantically equivalent completions, rather than a single surface form.

Let S denote an input sentence prefix (e.g., "this website is safe to "), and let T be the original content token ("browse"). We define T^* as the set of contextual synonyms for T {*browse, search, visit, surf, navigate, access, click, view, ...*}. Instead of optimizing the likelihood of predicting the exact token T, the concept objective rewards the model for predicting *any* token belonging to the conceptual equivalence class T^*. Formally:

$$\mathcal{L}_{\text{concept}}(T^* \mid S) = -\log \sum_{n=1}^{|T^*|} p(t_n \in T^* \mid S, \Theta)$$

where Θ denotes the model parameters and $p(t_n \in T^* \mid S, \Theta)$ is the probability assigned by the model to the n-th token in the contextual synonym set T^*. This objective **encourages the model to distribute probability mass across multiple realizations of the same concept**, rather than concentrating exclusively on a single lexical form.

3.3 Combined Training Loss

We combine the standard NTP loss with the concept loss using a weighted linear interpolation:

$$\mathcal{L}_{\text{total}} = (1 - \lambda)\, \mathcal{L}_{\text{NTP}} + \lambda\, \mathcal{L}_{\text{concept}}$$

where $\lambda \in [0, 1]$ controls the strength of concept-level supervision. Setting $\lambda = 0$ is equivalent to standard NTP training, while larger values place increasing emphasis on concept prediction.

To prevent the concept objective from dominating NTP, we introduce a probability mass threshold on the contextual synonym set. If the cumulative probability assigned by the model to tokens in T^* exceeds a certain threshold (we used 0.6), we set $\mathcal{L}_{\text{concept}} = 0$ for that training example. This ensures that once the model sufficiently captures the concept, additional optimization pressure is removed. In practice, we found that this threshold was only reached less than 1% of the time during training.

3.4 Models

We post-trained *Llama 3.1 8B*, *Llama 3.2 3B*, and *Llama 3.2 1B* (Grattafiori et al., 2024) using concept loss weights $\lambda \in \{0, 0.25, 0.5, 0.75, 1.0\}$, to evaluate the effect of increasing concept weight and model size (training details in Appendix B).

We evaluate our models against two baselines: the base model with no post-training and the base model post-trained with only NTP loss ($\lambda = 0$) using the same training data. The former measures the effect of post-training; the latter isolates the contribution of the concept objective.

We trained all model variants on C4 and OpenWebText separately to assess the effect of training data. Figure 2 depicts the datasets, models, and concept weights we used. In total, we post-trained 30 models using 3 pretrained base models.

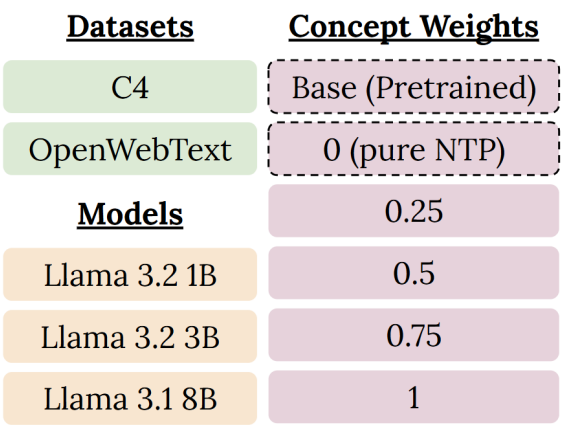


Figure 2: We post-trained using two datasets, three base models, and five concept weights. Baselines (base pretrained model with no training and concept weight 0 = pure NTP training) are in dashed black. All model variants with their performance are in Appendix H.

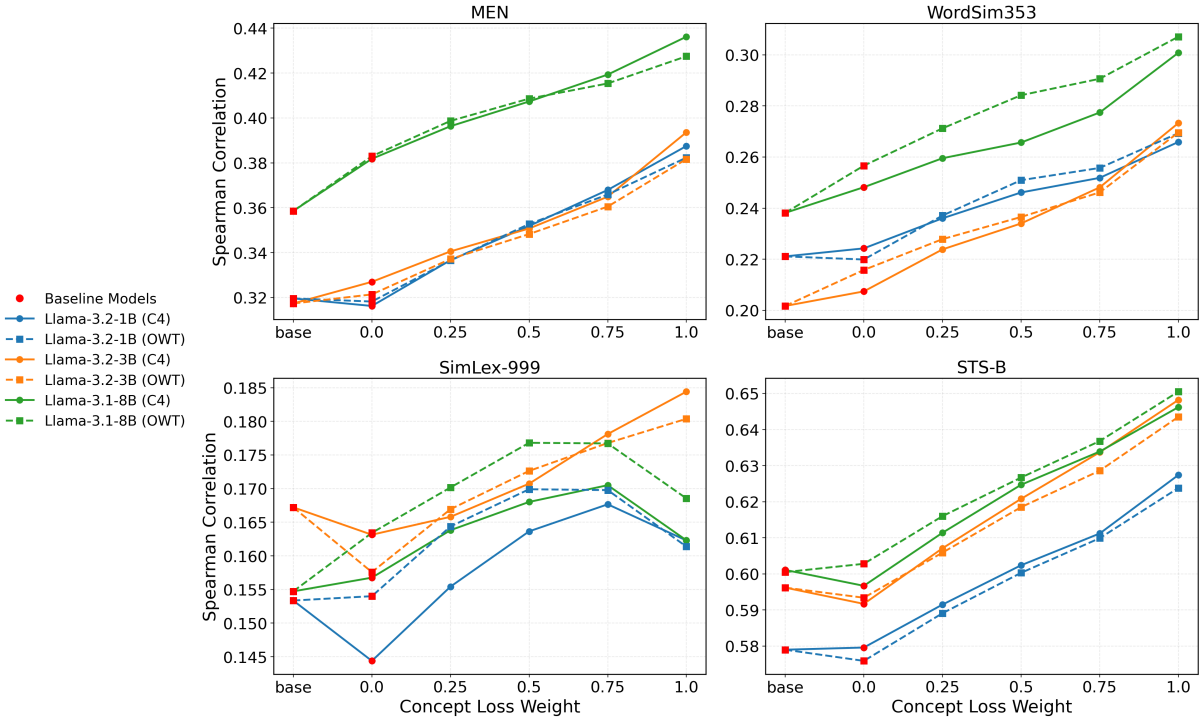


Figure 3: **Models with higher levels of concept awareness align more closely with human semantic intuition** on MEN, WordSim353, SimLex-999, and STS-B. Pure NTP post-training slightly degrades semantic understanding, while concept supervision improves alignment with human semantic judgments. For SimLex-999, concept training helps except at concept weight 1, suggesting token-level supervision is more important for this benchmark. Red dots indicate baseline models (pretrained and pure NTP post-trained models).

4 Experiments & Results

We evaluate the base, NTP, and concept models on alignment with human semantic judgments (Section 4.1) and NTP perplexity and accuracy on held-out data (Sections 4.2 & 4.3).

4.1 Increased Semantic Understanding

We evaluate alignment with human semantic judgments using both word pairs and sentence pairs with the following standard benchmarks:

MEN (Bruni et al., 2014) contains 3,000 word pairs annotated by 50 humans based on relatedness judgments. Participants were not instructed about the difference between similarity and relatedness, where similarity can refer to {"car, automobile"}) and relatedness can be {"wheels, car"}.

WordSim353 (Finkelstein et al., 2002) consists of 353 word pairs annotated by 13-16 humans. Participants estimated the relatedness of each pair on a scale from 0 to 10.

SimLex-999 (Hill et al., 2014) contains 999 word pairs. Unlike MEN and WordSim353, SimLex-999 quantifies similarity rather than association or relatedness. Similarity is defined as synonymy, and each pair is rated by roughly 50 humans.

STS-B (Cer et al., 2017) consists of 1,500 sentence pairs, drawn from news headlines, video and image captions, and natural language inference data. Each pair is human-annotated with a normalized similarity between 0 and 1.

For all benchmarks, we feed each word or sentence through the model and apply mean pooling over the final-layer hidden states, followed by L2 normalization. We then compute cosine similarity between the two vectors in each pair and report Spearman correlation with human judgments.

Concept-trained models achieve substantially higher Spearman correlation than both baselines on all semantic similarity tasks (Figure 3). In contrast, NTP-only post-training slightly degrades semantic alignment relative to the pretrained model, suggesting that continued token-level optimization alone can weaken semantic structure, whereas concept supervision mitigates this effect.

For the SimLex-999 benchmark, we note that the optimal concept weight is 0.75 rather than 1 for the 1B and 8B models. We observe the same general

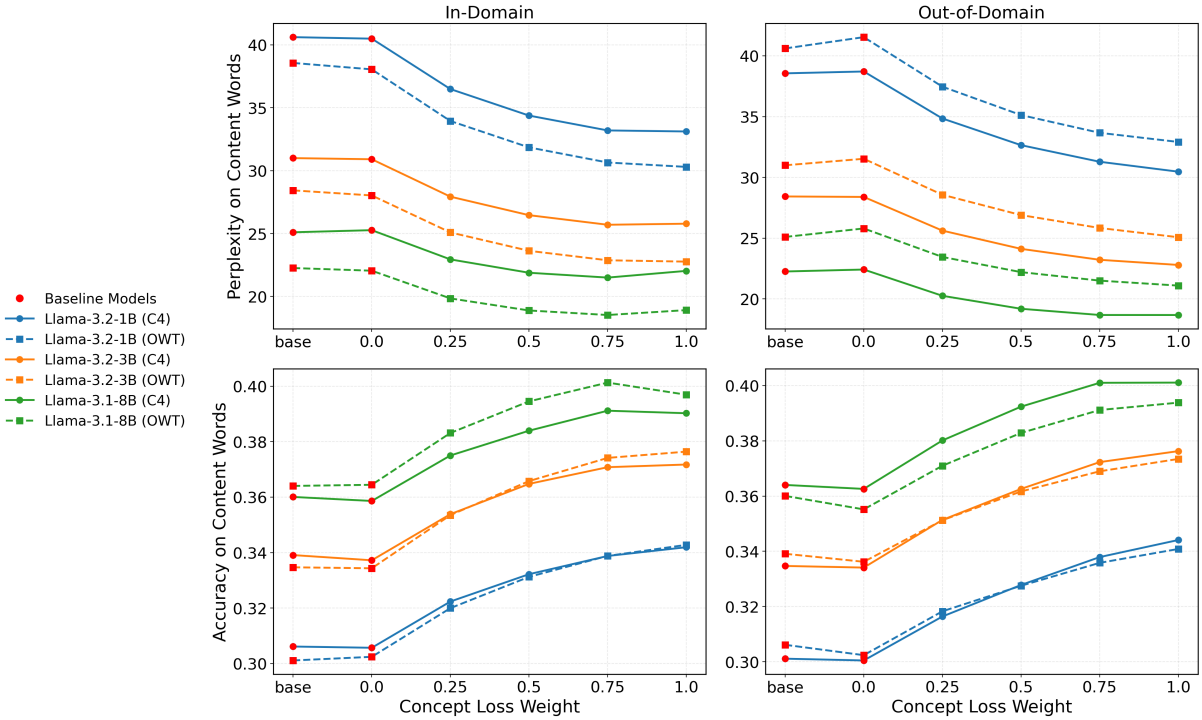


Figure 4: Content word NTP perplexity and accuracy using a held-out set of 1,000 C4 and OpenWebText (OWT) samples. The in-domain plots show results for models trained on one dataset and evaluated on the same dataset, while OOD means training on C4 and evaluating on OWT or vice versa (training data in legend). **Performance generally improves as concept supervision increases**, with some models showing a small dip between concept weights 0.75 and 1.0. Red dots indicate baseline models (pretrained models and the pure NTP post-trained models).

trend that concept training helps with human semantic alignment, but the decrease in correlation at concept weight 1 may indicate that token-level supervision is more important for SimLex-999 compared to MEN, WordSim353, and STS-B.

4.2 New Metric: Content Word Perplexity

While we deem human alignment to be the most relevant metric, this approach requires manual annotation and is therefore costly and difficult to scale. In contrast, perplexity remains the dominant automatic metric for LLM evaluation due to its simplicity and ease of computation.

Yet, standard perplexity treats all tokens equally, averaging across *function* words and *content-bearing* words. This can obscure meaningful improvements in semantic modeling. A growing body of work highlights that not all tokens contribute equally to learning or evaluation. Several studies show that prioritizing more informative or challenging tokens can improve model performance on downstream tasks (Bigelow et al., 2025; Thankaraj et al., 2025; Lin et al., 2024; Yang et al., 2025).

Consistent with this view, recent work has questioned the practice of computing perplexity by averaging across all tokens. For example, Fang et al. (2024) show that perplexity computed only on *key tokens* correlates better with performance on long-context benchmarks. Similarly, Miralles-González et al. (2026) argue that aggregating token-level metrics via simple averaging ignores differences in token predictive difficulty, and propose weighting schemes that emphasize more informative tokens.

Other work also highlights mismatches between global perplexity and model quality. Luitel et al. (2025) show that lower perplexity does not necessarily imply better performance, and Meister and Cotterell (2021) find that low-perplexity models may still fail to capture important linguistic properties. Together, these findings suggest that the **standard global perplexity approach, which treats all tokens equally, may obscure meaningful differences in token importance**.

Motivated by these observations, we introduce a more informative metric, which we call **content word perplexity**: standard NTP perplexity computed only over whole-word tokens that are nouns, verbs, or adjectives. These tokens typically carry most of the semantic content of a sentence, while less informative tokens such as “is”, “a”, “are”, and

"the", as well as subword fragments, are excluded.

We define content word perplexity as:

$$\text{Content Word PPL} = \exp\left(\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \text{NLL}(t)\right)$$

where $\mathcal{T}$ denotes the set of content word tokens in a sequence, and $\text{NLL}(t)$ is the negative log-likelihood of token t.

We evaluate perplexity and accuracy over content words rather than averaging across all tokens. Experiments are conducted on 1,000 held-out samples from each training corpus. We report results for both in-domain data (held-out samples from the training corpus) and out-of-distribution (OOD) data (samples from the other corpus).

We find that **concept models outperform the pretrained and NTP-trained baselines in content word accuracy and perplexity** (Figure 4), suggesting that concept-level supervision provides a richer learning signal than standard NTP training. Notably, NTP models ($\lambda = 0$) show no difference relative to the base model, whereas models with concept training exhibit lower perplexity and higher accuracy (significant under paired bootstrap confidence intervals; see Appendix C).

Concept models also demonstrate stronger OOD generalization. When models trained on C4 are evaluated on OpenWebText (and vice versa), they achieve lower perplexity and higher accuracy than both baselines (significant under paired bootstrap confidence intervals; see Appendix C). These findings suggest that concept-level supervision improves robustness to domain shift and leads to models that generalize more reliably to unseen data.

We also observe that increasing concept weight from $\lambda = 0$ to $\lambda = 0.75$ yields monotonic improvements in content word performance. At $\lambda = 1$, however, some models plateau or slightly degrade, suggesting that retaining some degree of token-level supervision is beneficial. In practice, **combining concept-level and token-level objectives yields the strongest overall performance**.

4.3 Global Perplexity

Figure 9 shows that **concept models exhibit minor degradation** relative to the base model for both in-domain and OOD evaluation, with the $\lambda = 1$ model showing the largest drop. This pattern is consistent with our objective: concept training emphasizes semantically meaningful tokens, which are typically harder to predict than the function words that dominate global perplexity (content words make up 28% of tokens in our data). For $\lambda = 1$, the larger degradation likely reflects that the model is trained on fewer tokens, since loss is applied only to content words, while the other models receive NTP supervision on all tokens.

We conclude that models with concept weights of 0.25 and 0.5 achieve the best trade-off, **improving human alignment while increasing global perplexity by less than one point and reducing content-word perplexity by more than five points** (Appendix D).

5 Analysis and Ablations

We further analyze our concept training manipulation. Specifically, we examine how it affects latent space geometry (Section 5.1), how it interacts with model size (Section 5.2), the effect of introducing noise (Section 5.3), and the impact of reducing the number of content words in the data (Section 5.4).

5.1 Improved Cluster Separation

To examine the effect of concept training on internal geometry, we analyze hidden states on 1,000 held-out samples. For each sequence, we randomly select one content word and replace it with each synonym in its concept set to construct concept variations. We then extract the last-layer hidden state of the content token for all tokens in the set.

We evaluate whether synonyms for the same concept occupy a coherent region in representation space using intra- and inter-concept cosine similarity. **Intra-concept similarity** is the mean pairwise cosine similarity among hidden states sharing a concept, while **inter-concept similarity** is the mean across words belonging to *different* concepts. The clustering score is the difference between these values. We find that all models have comparable intra-concept similarity, but concept models show notably lower inter-concept similarity, indicating that concept loss sharpens decision boundaries rather than collapsing embeddings (Figure 5, left). Thus, concept supervision combined with NTP training preserves representational expressivity while learning more meaningful semantic boundaries.

We observe that the 8B model shows smaller clustering improvements than the 1B and 3B models, likely because the base 8B model already exhibits stronger semantic understanding and therefore benefits less from additional concept learning.

We also compute each concept's centroid and

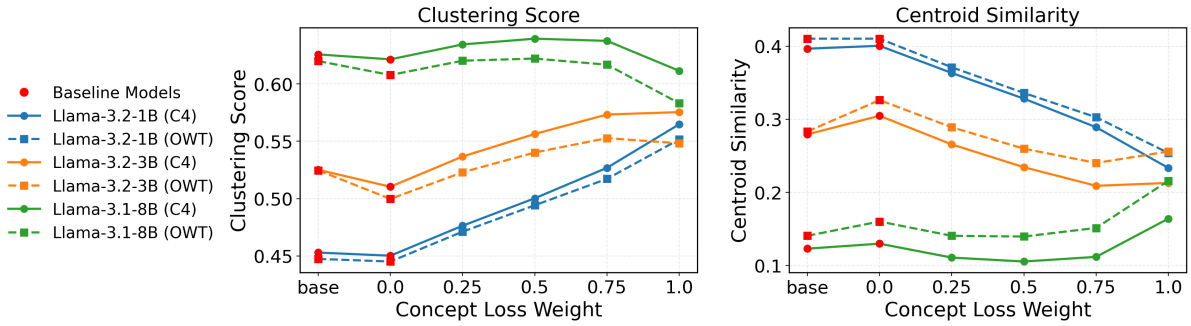


Figure 5: **Clustering ability generally improves as concept weight increases**, with degradation at concept weight 1. These results show that concept training helps models to learn better semantic distinctions, but some amount of NTP supervision is necessary to maintain general semantic understanding. Red dots indicate baseline models.

report the mean pairwise cosine similarity across centroids. Centroid similarity decreases as concept weight increases, consistent with the inter-concept separation above (Figure 5, right).

5.2 Effect of Model Size

We examine how model size affects concept training. First, we see a clear scale effect on content word NTP performance (Figure 4), and a more noisy effect for alignment with semantic similarity (much clearer without concept training; Figure 3).

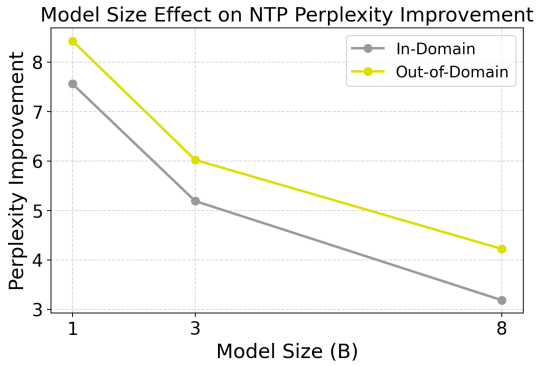


Figure 6: NTP perplexity improvement for concept weight = 0 versus 1 across model sizes. **Smaller models benefit more from concept supervision**, showing larger perplexity reductions than larger models.

Figure 6 compares the improvement in NTP perplexity between concept weight 0 and 1, showing that smaller models gain more from concept supervision compared to bigger models, consistent with our observation in Section 5.1. For alignment with semantic similarity, we see a similar but much weaker trend (see Figure 10 in Appendix E). These results suggest that concept supervision is most helpful for smaller models, where encouraging abstraction at the concept level may help allocate limited representational capacity more efficiently.

5.3 Learning from Noise

To verify that gains stem from semantically coherent synonym sets rather than mere lexical variation (i.e., noise as regularization), we train control models with randomized synonym assignments. We pool all synonyms from the training set and, for each sequence, randomly sample a matching number of words from this pool, replacing every true synonym set with this noise-based control.

As representatives of the general trend, we show the in-domain NTP accuracy on content words and the WordSim353 correlation in Figure 7 (all results in Appendix F and Appendix H). We find that even at the weight on the noise "concept" loss, models trained with randomized synonym assignments perform substantially worse than both the pretrained and NTP-trained baselines on both NTP and human semantic alignment benchmarks. These results indicate that the **improvements from concept training are not simply due to additional label variability or regularization**. Instead, semantically structured concept supervision appears necessary for the observed gains, enabling models to improve performance on both NTP and human alignment tasks relative to all baselines.

5.4 Supervision Proportion

We trained our models on all content words in our data. To assess the effect of different amounts of content words used for training, we run our concept training on different proportions of content words from each sequence. We evaluate concept training using 100%, 50%, 25%, and only the last of these content words. For the last content word, we use the last word in each sequence, where we filtered sequences to end in a content word.

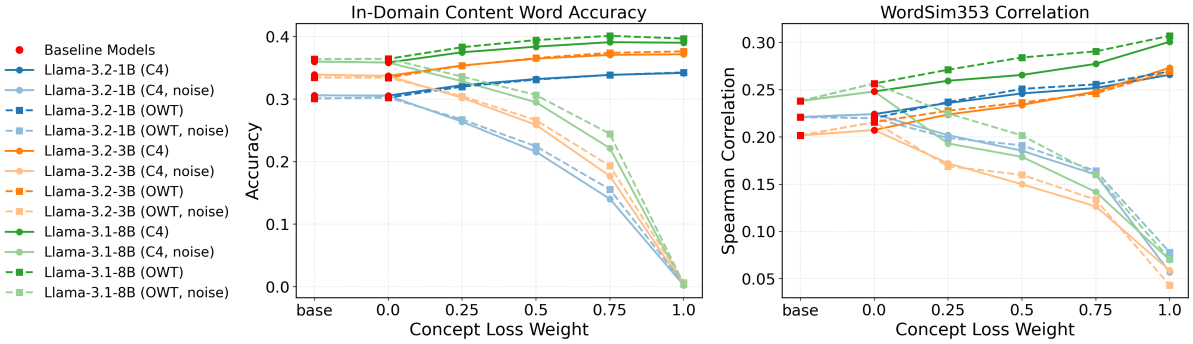


Figure 7: Noise models perform substantially worse than models with semantically structured clusters, with monotonically decreasing accuracy and human similarity correlation as more noise is injected. Red dots indicate baseline models (pretrained models and the concept weight 0 pure NTP post-trained models).

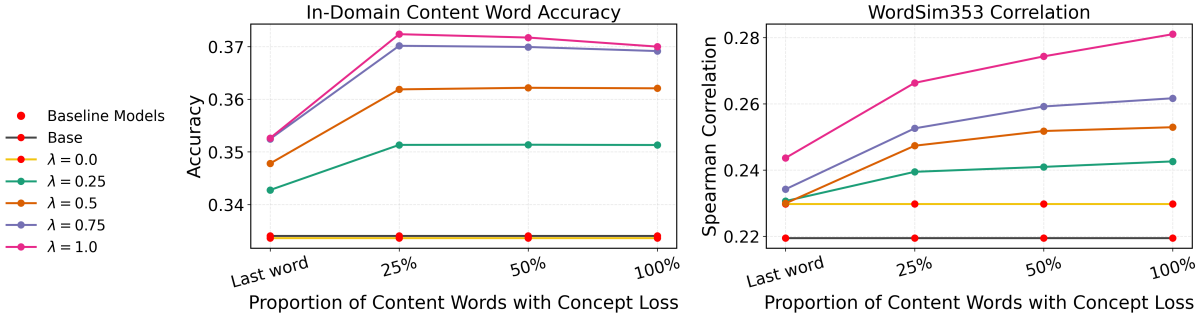


Figure 8: As the proportion of content words receiving concept loss increases, content word accuracy initially increases and then remains stable from 25% to 100%. Alignment with human semantic judgments on WordSim353 steadily improves. Results are averaged across model sizes and datasets.

Similar to Section 5.3, we show representative results for NTP content word accuracy and WordSim353 correlation in Figure 8. Even with concept training on only the last word in each sequence, we observe notable improvements in human semantic alignment, and increasing the proportion of concept supervision further improves performance. Content word perplexity and accuracy remain similar for models with 25%, 50%, and 100% supervision, suggesting that 25% supervision is sufficient to saturate NTP performance, while higher proportions primarily improve alignment with human semantic judgment. These results suggest that NTP metrics may saturate before models fully capture conceptual structure, while human-aligned semantic benchmarks provide a more sensitive measure of improvements in semantic understanding. Full results for this ablation are reported in Appendix G.

6 Conclusions and Future Directions

We introduce a self-supervised method for training LLMs using a concept-based objective that complements standard NTP. We show that concept-level models learn more meaningful semantic boundaries, align better with human intuition, and better predict semantically informative tokens in both in-distribution and OOD settings. These results suggest that incorporating concept signal into training can yield more human-like representations.

Our results also highlight the importance of the evaluation metric. Concept models better model content words, but not all other tokens. Because frequent function words dominate global perplexity, improvements on meaningful tokens can be obscured. Evaluating content word perplexity better reflects how models capture semantic structure.

Finally, we find that **combining concept-level and token-level supervision yields the strongest overall performance**, suggesting that concept objectives can serve as a complementary training signal rather than a replacement for NTP.

Future work can explore multilingual data, multi-token expressions, and evaluations on additional downstream tasks. More broadly, we suggest that incorporating concept-level training objectives and evaluating models on semantically meaningful predictions can lead to LLMs that better reflect the conceptual organization underlying human language.

Limitations

Although we show promising results with concept training, our study has several limitations. First, our experiments focus exclusively on decoder-based models, which limits our ability to evaluate performance under instruction prompting for downstream tasks. As a result, our evaluation tasks in this work are relatively small in scope and do not capture ways in which concept training could be applied to improve downstream performance.

Second, our concept supervision is applied only to nouns, verbs, and adjectives that are a single token and a complete word. This restricts our training paradigm to a narrow prediction setting and may not fully capture the benefits of concept-level learning across longer and more diverse spans of text. Extending supervision to additional tokens within sequences could yield stronger effects.

Third, our method for constructing concept sets relies on LLM-generated synonym clusters. While scalable, this approach may not perfectly reflect human semantic organization, and alternative sources, such as human annotations, could provide more reliable or linguistically grounded concept groupings.

Finally, our concept weight 1 model is trained on fewer tokens than all other models (as it is only trained on content words and the other models have some degree of NTP on non-content tokens), which might account for its degradation in global NTP performance and SimLex-999 correlation.

Ethical Considerations

Training models with our concept loss formulation could potentially lead to overgeneralizing and overextending concept boundaries. The approach we describe in this proposal involves LLMs creating concept sets without human input, which can be dangerous in amplifying biases and stereotypes present in training data. We advise users to be careful of potentially harmful associations generated in the concept dataset creation.

As with NTP models, concept-trained models may behave in an unwanted or harmful manner, such as producing hallucinations. In this work, we did not explore using our concept-trained models for unconditioned generation, and thus the severity of these risks is unknown for our models. We caution users to always check model outputs.

Disclosure: LLMs were used to assist with coding and refine the text of this paper.

References

Kwangjun Ahn, Alex Lamb, and John Langford. 2025. Efficient joint prediction of multiple future tokens. *arXiv preprint arXiv:2503.21801*.

Gregor Bachmann and Vaishnavh Nagarajan. 2024. The pitfalls of next-token prediction. In *Forty-first International Conference on Machine Learning*.

Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, Mariano Coria, Guillaume Couairon, Marta R Costa-jussà, David Dale, and 1 others. 2024. Large concept models: Language modeling in a sentence representation space. *arXiv preprint arXiv:2412.08821*.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Eric J Bigelow, Ari Holtzman, Hidenori Tanaka, and Tomer Ullman. 2025. Forking paths in neural text generation. In *The Thirteenth International Conference on Learning Representations*.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, page 1–14. Association for Computational Linguistics.

Michael K Chen, Xikun Zhang, Jiaxing Huang, and Dacheng Tao. 2025. Improving large language models with concept-aware fine-tuning. *arXiv preprint arXiv:2506.07833*.

Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, Stefanie Jegelka, Jinyang Gao, Bolin Ding, and Yisen Wang. 2024. What is wrong with perplexity for long-context language modeling? *arXiv preprint arXiv:2410.23771*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Preprint*, arXiv:1408.3456.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Hai Huang, Yann LeCun, and Randall Balestriero. 2026. LLM-JEPA: Large language models meet joint embedding predictive architectures. In *The Fourteenth International Conference on Learning Representations*.

Laya Iyer, Pranav Somani, Alice Guo, Dan Jurafsky, and Chen Shani. 2026. Beyond tokens: Concept-level training objectives for llms. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics*.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. Sensebert: Driving some sense into bert. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2024. Not all tokens are what you need for pretraining. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Tao Liu, Taiqiang Wu, Runming Yang, Shaoning Sun, Junjie Wang, and Yujiu Yang. 2026. Profit: Leveraging high-value signals in sft via probability-guided token selection. *arXiv preprint arXiv:2601.09195*.

Nishant Luitel, Nirajan Bekoju, Anand Kumar Sah, and Subarna Shakya. 2025. Can perplexity predict fine-tuning performance? an investigation of tokenization effects on sequential language models for nepali. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM^2)*, pages 239–248.

Divyat Mahajan, Sachin Goyal, Badr Youbi Idrissi, Mohammad Pezeshki, Ioannis Mitliagkas, David Lopez-Paz, and Kartik Ahuja. 2026. Beyond multi-token prediction: Pretraining llms with future summaries. In *The Fourteenth International Conference on Learning Representations*.

Clara Meister and Ryan Cotterell. 2021. Language model evaluation beyond perplexity. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 5328–5339.

William Merrill, Zhaofeng Wu, Norihito Naka, Yoon Kim, and Tal Linzen. 2024. Can you learn semantics through next-word prediction? the case of entailment. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2752–2773.

Pablo Miralles-González, Javier Huertas-Tato, Alejandro Martín, and David Camacho. 2026. Not all tokens are created equal: Perplexity attention weighted networks for ai-generated text detection. *Information Fusion*, 125:103465.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Alexander Shabalin, Viacheslav Meshchaninov, Egor Chimbulatov, Vladislav Lapikov, Roman Kim, Grigory Bartosh, Dmitry Molchanov, Sergey Markov, and Dmitry Vetrov. 2025. Tencdm: Understanding the properties of the diffusion model in the space of language model encodings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25110–25118.

Chen Shani, Liron Soffer, Dan Jurafsky, Yann LeCun, and Ravid Shwartz-Ziv. 2026. From tokens to thoughts: How llms and humans trade compression for meaning. In *The Fourteenth International Conference on Learning Representations*.

Chen Shani, Jilles Vreeken, and Dafna Shahaf. 2023. Towards concept-aware large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13158–13170.

Jihoon Tack, Jack Lanchantin, Jane Yu, Andrew Cohen, Ilia Kulikov, Janice Lan, Shibo Hao, Yuandong Tian, Jason E Weston, and Xian Li. 2026. LLM pretraining with continuous concepts. In *The Fourteenth International Conference on Learning Representations*.

Abitha Thankaraj, Yiding Jiang, J Zico Kolter, and Yonatan Bisk. 2025. Looking beyond the next token. *arXiv preprint arXiv:2504.11336*.

Hui Wei, Zihao Zhang, Shenghua He, Tian Xia, Shijia Pan, and Fei Liu. 2025. Plangenllms: A modern survey of llm planning capabilities. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19497–19521.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36:59008–59033.

Charlie Wyatt, Aditya Joshi, and Flora Salim. 2025. Alternatives to next token prediction in text generation–a survey. *arXiv preprint arXiv:2509.24435*.

Tianyi Yan, Fei Wang, James Y Huang, Wenxuan Zhou, Fan Yin, Aram Galstyan, Wenpeng Yin, and Muhao Chen. 2024. Contrastive instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10288–10302.

Chun-Hao Yang, Bo-Han Feng, Tzu-Yuan Lai, Yan Yu Chen, Yin-Kai Dean Huang, and Shou-De Lin. 2025. Training llms beyond next token prediction–filling the mutual information gap. *arXiv preprint arXiv:2511.00198*.

A Prompt to Extract Concept Sets

This is the prompt for Llama 3.1 8B Instruct to select the subset of contextual synonyms from the top 200 decoded tokens generated by Llama 3.1 8B. We use deterministic sampling for reproducibility and a repetition penalty of 1.1.

```
Find contextual synonyms for the word
{target} in this text: {input_sequence}

Available tokens: {decoded_tokens}

Instructions:

- Find ALL possible synonyms from the
available tokens that could replace
{target} in this context

- Return ONLY a comma-separated list
of synonyms from the available tokens,
surrounded by square brackets

- Include every relevant synonym

- NO duplicates allowed

- NO explanations or extra text

- If no synonyms found, return: []

Example format: [word1, word2, word3]

Synonyms for {target}:
```

B Training Details

We split each dataset into 90% train and 10% validation. All models are trained using LoRa (Hu et al., 2022) with a learning rate of 7×10^{-5} and batch size of 2 for 5 epochs, leading to a training time of approximately 1.5 hours on an NVIDIA RTX A6000 GPU. The loss plateaus by the end of the fifth epoch — with early stopping enabled, we find that models typically stop training around 4 epochs. We limit the contextual synonym set size to a maximum of 10 excluding the content word itself, and we use a probability mass threshold of 0.6.

C Statistical Significance Test for Content Word NTP Improvements

We report our results for paired bootstrap confidence intervals for the difference in content word negative log likelihood (NLL) and accuracy between each concept-trained model and the pretrained base model. These metrics are computed on content words in 1,000 held-out samples. We divide our results into two tables, with Table 1 for in-domain results and Table 2 for out-of-domain results. We use OWT as an abbreviation for OpenWebText.

We conduct our bootstrapping with 1,000 bootstrap samples and a 95% confidence interval.

D Global Perplexity

In Figure 9, we report the global perplexity for our models, which includes all tokens in each sequence.

E Effect of Model Size on Semantic Similarity Benchmarks

Similar to Figure 6, we evaluate the effect of model size for the human semantic alignment benchmarks MEN, WordSim353, SimLex-999, and STS-B. We show this result in Figure 10.

F Plots Including Noise Models

As described in Section 5.3, we include plots showing models trained with randomized synonym assignments, along with their counterparts trained using proper synonyms. Figure 11 shows alignment with human semantic judgments on the benchmarks that we use in the main text. Figure 12 illustrates their performance on NTP perplexity and accuracy for content words. Figure 13 demonstrates the clustering geometry results.

G Plots Comparing Different Supervision Proportions of Content Words

As described in Section 5.4, we report results for models trained on different amounts of content words within each sequence. We run ablations for only the last content word, 25% of content words, 50%, and 100%. In the main paper, we primarily discuss the 100% model.

For simplicity in the plots below, we average over the three different model sizes (1B, 3B, and 8B) as well as the two datasets (C4 and OpenWebText). Figure 14 shows results on the human semantic similarity tasks for different proportions of concept supervision. Figure 15 shows results on perplexity and accuracy for content words, and Figure 16 illustrates the clustering results.

H Numerical Results for 100% Proportion Model

We include all numerical results for the models trained using 100% of the possible content words in each sequence, as these are the models that we primarily discuss in the main text. Tables 3 and 4 show results for the models trained on C4: the former reports the human semantic alignment benchmarks, and the latter reports the NTP and clustering metrics. Tables 5 and 6 follow the same format for the models trained on OpenWebText (OWT).

Model	Train/Eval	λ	Accuracy ↑ Lower	Accuracy ↑ Upper	NLL ↓ Lower	NLL ↓ Upper
Llama 3.2 1B	C4 / C4	0.0	-0.0034	-0.0009	-0.0047	-0.0012
		0.25	$+0.0136$	$+0.0171$	-0.1082	-0.1028
		0.5	$+0.0235$	$+0.0277$	-0.1671	-0.1590
		0.75	$+0.0304$	$+0.0355$	-0.2026	-0.1916
		1.0	$+0.0313$	$+0.0370$	-0.2065	-0.1909
	OWT / OWT	0.0	$+0.0002$	$+0.0024$	-0.0154	-0.0113
		0.25	$+0.0174$	$+0.0204$	-0.1303	-0.1251
		0.5	$+0.0283$	$+0.0320$	-0.1951	-0.1878
		0.75	$+0.0356$	$+0.0398$	-0.2347	-0.2250
		1.0	$+0.0393$	$+0.0441$	-0.2480	-0.2343
Llama 3.2 3B	C4 / C4	0.0	-0.0014^*	$+0.0006^*$	-0.0026	-0.0001
		0.25	$+0.0158$	$+0.0187$	-0.1073	-0.1032
		0.5	$+0.0267$	$+0.0303$	-0.1698	-0.1633
		0.75	$+0.0350$	$+0.0395$	-0.2131	-0.2043
		1.0	$+0.0396$	$+0.0448$	-0.2386	-0.2261
	OWT / OWT	0.0	-0.0015^*	$+0.0009^*$	-0.0165	-0.0120
		0.25	$+0.0172$	$+0.0205$	-0.1281	-0.1222
		0.5	$+0.0290$	$+0.0332$	-0.1890	-0.1812
		0.75	$+0.0371$	$+0.0419$	-0.2226	-0.2122
		1.0	$+0.0390$	$+0.0444$	-0.2290	-0.2149
Llama 3.1 8B	C4 / C4	0.0	-0.0024	-0.0003	$+0.0054$	$+0.0085$
		0.25	$+0.0152$	$+0.0181$	-0.0988	-0.0945
		0.5	$+0.0265$	$+0.0301$	-0.1588	-0.1522
		0.75	$+0.0352$	$+0.0396$	-0.1961	-0.1869
		1.0	$+0.0359$	$+0.0411$	-0.2122	-0.1998
	OWT / OWT	0.0	-0.0009^*	$+0.0016^*$	-0.0117	-0.0072
		0.25	$+0.0174$	$+0.0207$	-0.1177	-0.1121
		0.5	$+0.0286$	$+0.0326$	-0.1683	-0.1605
		0.75	$+0.0348$	$+0.0395$	-0.1891	-0.1785
		1.0	$+0.0299$	$+0.0357$	-0.1703	-0.1554

Table 1: Paired bootstrap 95% confidence intervals of difference vs. base model (adapter − base), **in-domain** evaluation. Intervals marked * contain 0 and are not statistically significant. We see that the models trained with $\lambda = 0$ sometimes do not have significant differences compared to the pretrained model, and when they do, the difference is often in the undesired direction, indicating that pure NTP post-training may degrade model performance on content words. In contrast, models trained with concept supervision have notable improvements in both accuracy and NLL for content words.

Model	Train/Eval	λ	Accuracy ↑ Lower	Accuracy ↑ Upper	NLL ↓ Lower	NLL ↓ Upper
Llama 3.2 1B	C4 / OWT	0.0	-0.0016^*	$+0.0003^*$	$+0.0027$	$+0.0051$
		0.25	$+0.0139$	$+0.0167$	-0.1034	-0.0994
		0.5	$+0.0249$	$+0.0284$	-0.1695	-0.1634
		0.75	$+0.0347$	$+0.0388$	-0.2134	-0.2049
		1.0	$+0.0405$	$+0.0453$	-0.2415	-0.2294
	OWT / C4	0.0	-0.0051	-0.0025	$+0.0202$	$+0.0249$
		0.25	$+0.0105$	$+0.0139$	-0.0837	-0.0778
		0.5	$+0.0194$	$+0.0235$	-0.1489	-0.1409
		0.75	$+0.0271$	$+0.0321$	-0.1929	-0.1819
		1.0	$+0.0318$	$+0.0375$	-0.2180	-0.2025
Llama 3.2 3B	C4 / OWT	0.0	-0.0016^*	$+0.0005^*$	-0.0027	-0.0001
		0.25	$+0.0152$	$+0.0181$	-0.1061	-0.1020
		0.5	$+0.0260$	$+0.0299$	-0.1677	-0.1611
		0.75	$+0.0354$	$+0.0399$	-0.2071	-0.1977
		1.0	$+0.0390$	$+0.0442$	-0.2272	-0.2144
	OWT / C4	0.0	-0.0042	-0.0016	$+0.0148$	$+0.0195$
		0.25	$+0.0105$	$+0.0139$	-0.0844	-0.0785
		0.5	$+0.0204$	$+0.0247$	-0.1459	-0.1375
		0.75	$+0.0274$	$+0.0324$	-0.1878	-0.1765
		1.0	$+0.0315$	$+0.0373$	-0.2192	-0.2041
Llama 3.1 8B	C4 / OWT	0.0	-0.0026	-0.0003	$+0.0056$	$+0.0087$
		0.25	$+0.0145$	$+0.0176$	-0.0964	-0.0920
		0.5	$+0.0264$	$+0.0301$	-0.1519	-0.1449
		0.75	$+0.0348$	$+0.0392$	-0.1803	-0.1708
		1.0	$+0.0344$	$+0.0397$	-0.1823	-0.1691
	OWT / C4	0.0	-0.0062	-0.0034	$+0.0243$	$+0.0300$
		0.25	$+0.0091$	$+0.0127$	-0.0712	-0.0645
		0.5	$+0.0205$	$+0.0250$	-0.1268	-0.1181
		0.75	$+0.0288$	$+0.0336$	-0.1600	-0.1488
		1.0	$+0.0308$	$+0.0369$	-0.1817	-0.1658

Table 2: Paired bootstrap 95% confidence intervals of difference vs. base model (adapter − base), **out-of-domain** evaluation. Intervals marked * contain 0 and are not statistically significant. We see the same trends as in the in-domain evaluation. For cross-domain, the models trained with $\lambda = 0$ commonly perform worse than the pretrained models, again suggesting that pure NTP post-training may degrade model performance on content words. In contrast, models trained with concept supervision have notable improvements in both accuracy and NLL for content words across the board.

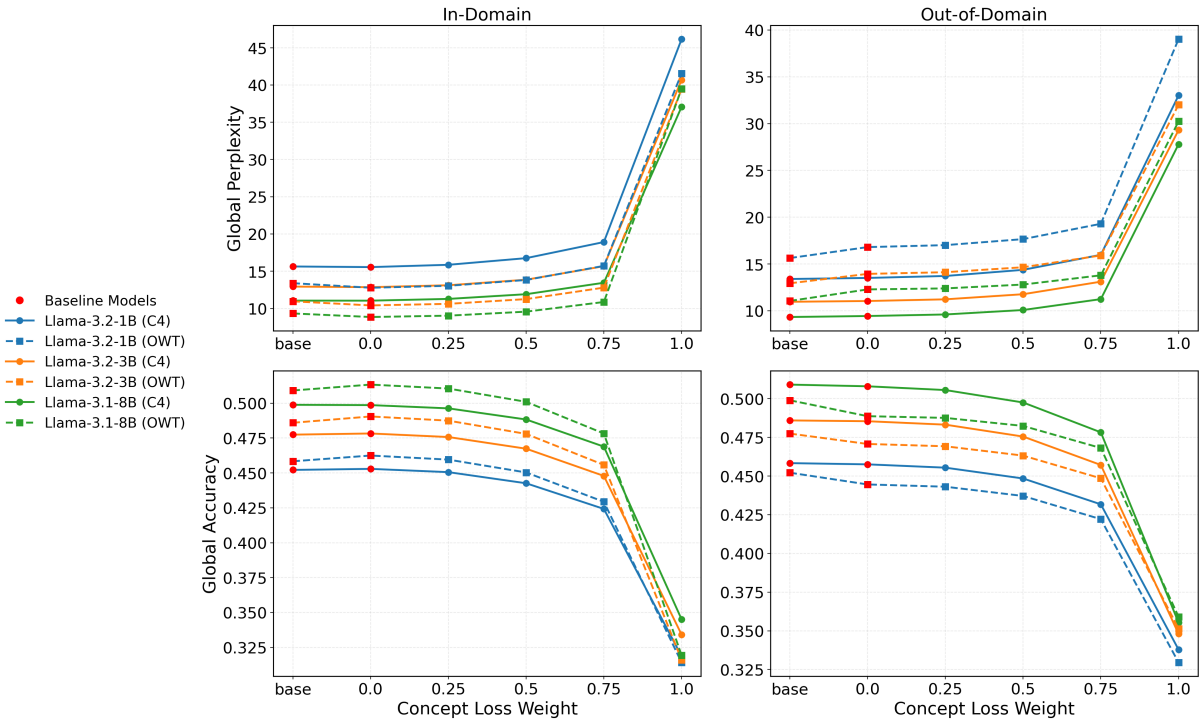


Figure 9: Global perplexity and accuracy across all tokens. For models with concept weights of 0.25 and 0.5, the increase in perplexity and corresponding decrease in accuracy are relatively small. We argue that this modest degradation is a worthwhile tradeoff given the substantially larger improvements in next-token prediction for content words. As shown in Figure 4, content words exhibit much lower accuracy and higher perplexity than the global averages reported here, indicating that they are generally more difficult for models to predict. Our models with weights 0.25 and 0.5 significantly improve performance on these challenging tokens while only slightly reducing global NTP performance. For the $\lambda = 1$ model specifically, we note that the large degradation is likely due to training on fewer tokens than the other models. This pure concept-trained model only has loss acting on the content words in each sequence, while the other models can learn from all tokens through NTP.

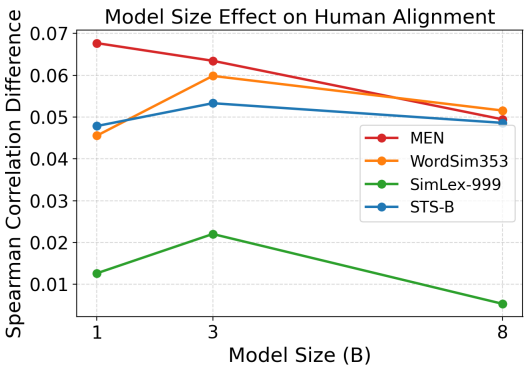


Figure 10: Spearman correlation difference on human alignment benchmarks for concept weight = 0 versus 1 across model sizes. Smaller models generally benefit more from concept supervision, showing larger Spearman correlation differences than larger models. This trend is clearest for MEN and SimLex-999. The 3B model experiences the largest improvement for WordSim353, and the differences are approximately the same across model sizes for STS-B.

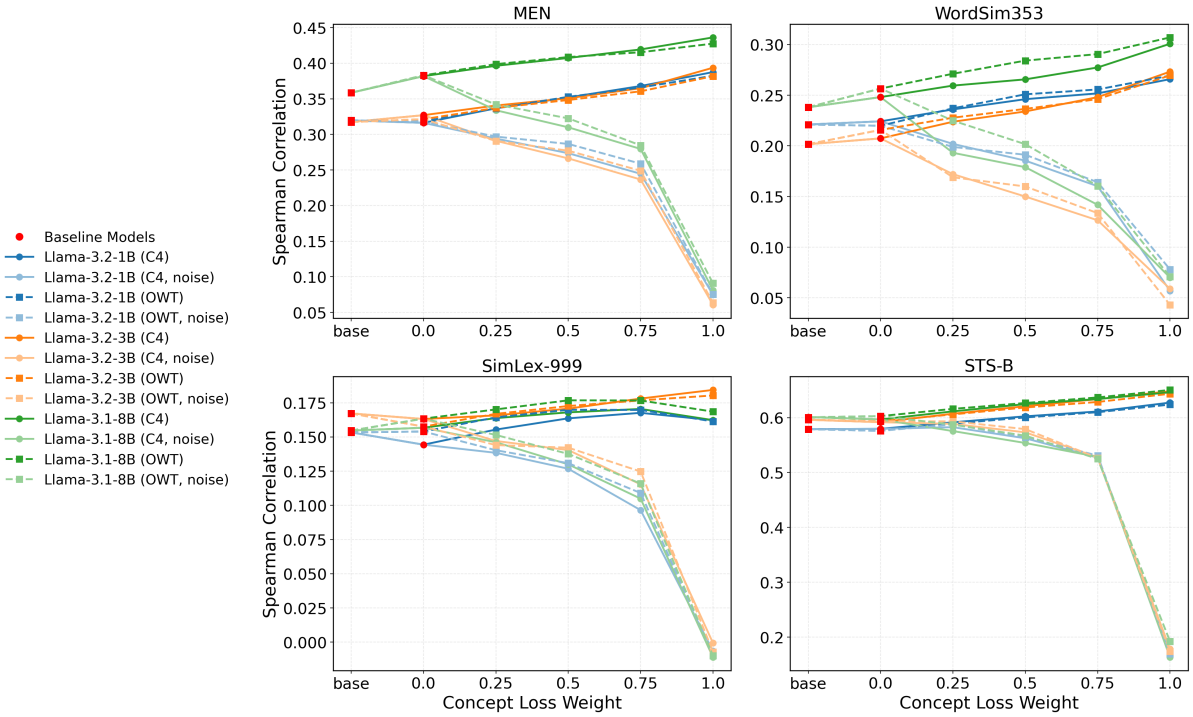


Figure 11: Correlation with human semantic judgments on MEN, WordSim353, SimLex-999, and STS-B benchmarks, along with results from models trained with randomized synonym assignments. We observe that training with random synonyms substantially degrades models' semantic understanding ability, making them significantly less aligned with human semantic intuition.

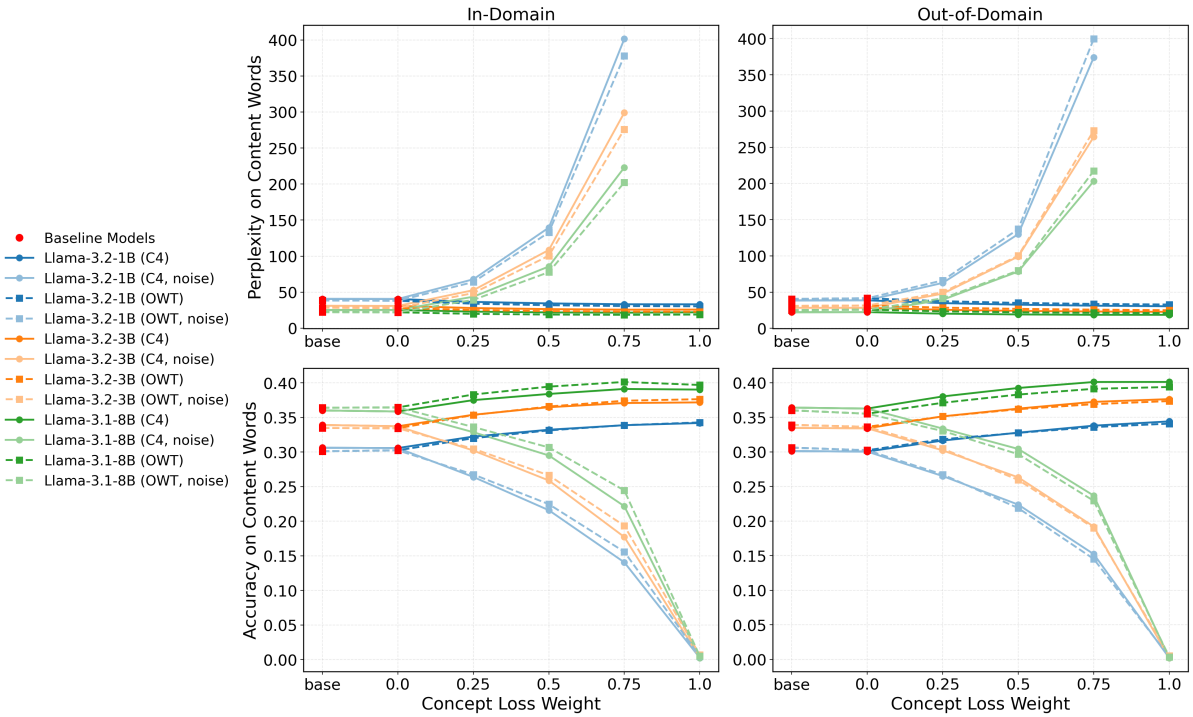


Figure 12: Perplexity and accuracy on content words including models trained with randomized synonym assignments. The weight 1 randomized models' perplexities are on the order of 10^4; we exclude them from the graphs so that they do not distort the perplexity plots' scales. We see that training with random synonyms significantly worsens both perplexity and accuracy, while training with our concept sets allows models to improve their NTP performance over the baselines.

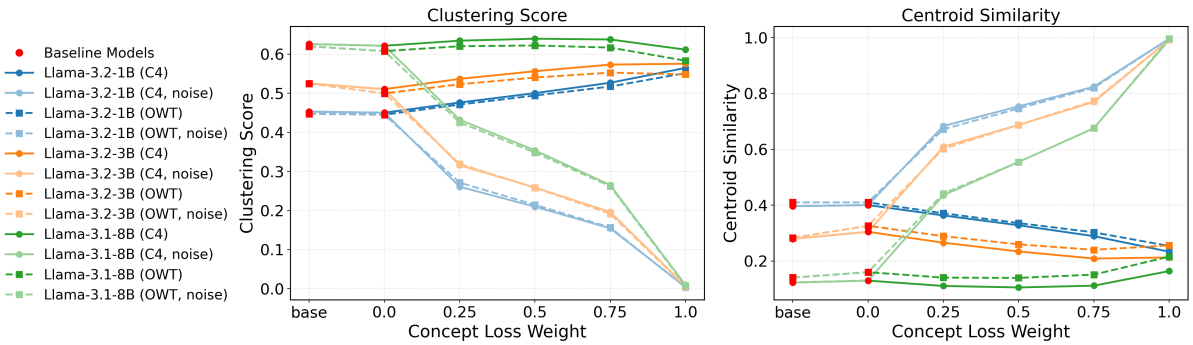


Figure 13: Clustering score plots including models trained with randomized synonym assignments. In agreement with Figure 11, training with random synonyms substantially degrades the models' ability to separate concept clusters, indicating that semantically coherent concept sets are necessary for learning meaningful semantic boundaries and that clustering improvements are not simply due to additional label variability during training.

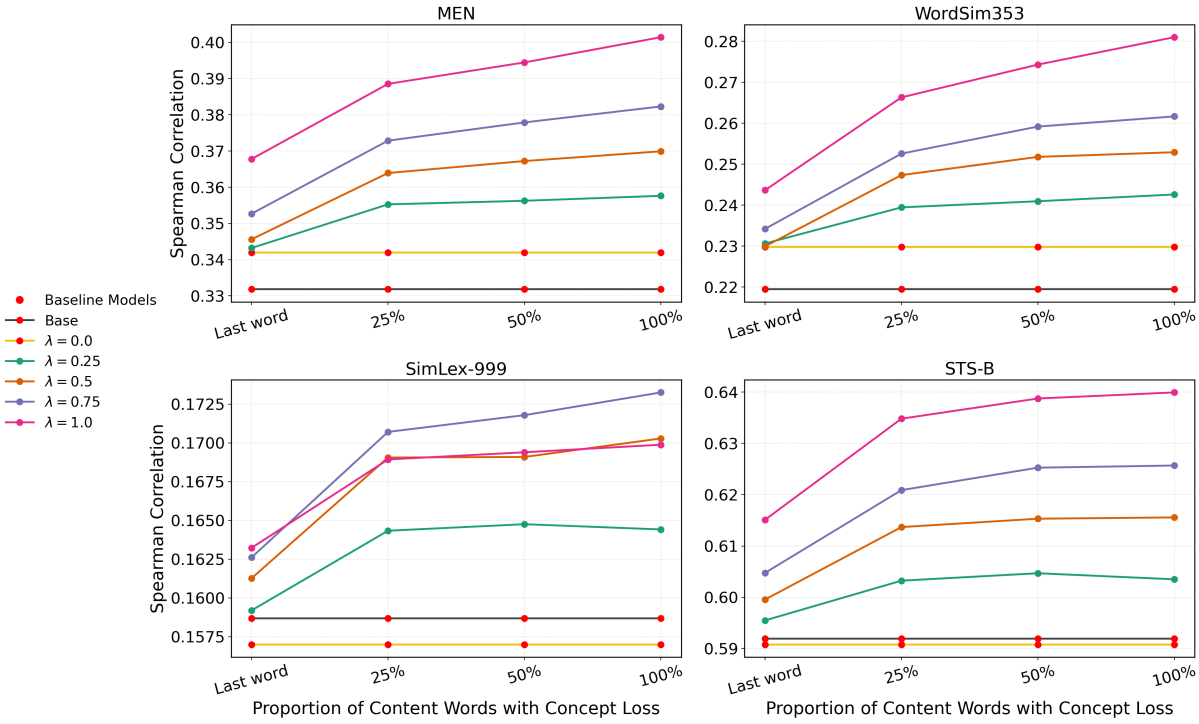


Figure 14: Correlation with human semantic judgments on MEN, WordSim353, SimLex-999, and STS-B benchmarks with results from models trained with different proportions of concept supervision, based on the number of content words participating in the concept loss. In general, increasing the proportion leads to stronger alignment with human semantic judgments, suggesting that concept-level supervision improves the semantic structure of learned representations.

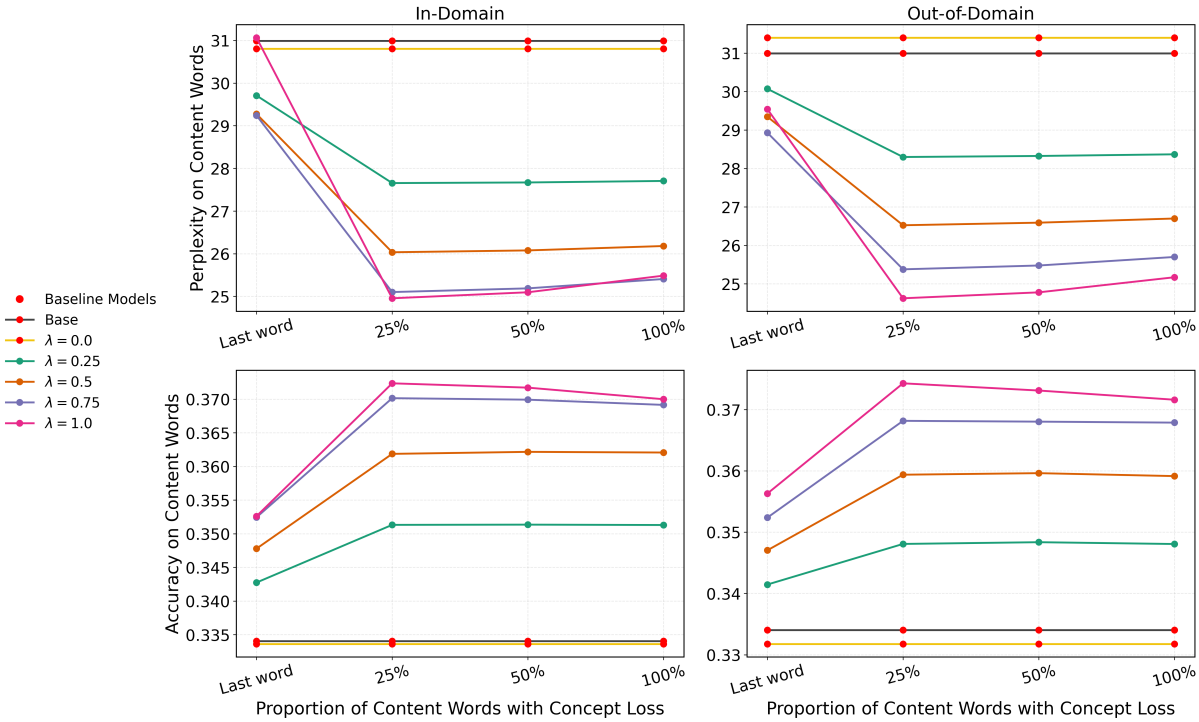


Figure 15: Perplexity and accuracy on content words for models trained with different proportions of content words. We observe a substantial improvement in both perplexity and accuracy when moving from training only on the final word to training on 25% of content words. However, there is little additional improvement between the 25%, 50%, and 100% settings. This suggests that concept training improves next-token prediction for content words, but increasing the proportion of such tokens used for supervision yields diminishing returns.

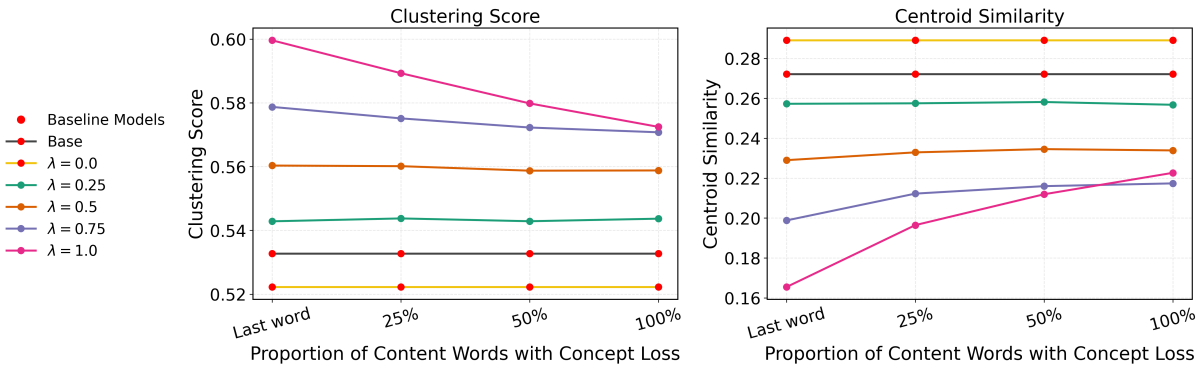


Figure 16: Clustering score and concept centroid similarity for models trained with different proportions of content words. For concept weights of 0.25, 0.5, and 0.75, clustering performance remains largely stable across all proportions. In contrast, the model trained with concept weight 1 shows degraded clustering as the proportion of tokens receiving concept loss increases. This suggests that relying exclusively on concept-level supervision ($\lambda = 1$) may compress representations excessively, as the model lacks the finer-grained signal provided by the standard NTP objective.

Model	Post-train Method	λ	MEN ↑	SimLex ↑	WordSim353 ↑	STS-B ↑
Llama 3.2 1B	Baseline	Base	0.3196	0.1534	0.2211	0.5790
		0.0	0.3162	0.1444	0.2242	0.5796
	Concepts	0.25	0.3366	0.1554	0.2360	0.5915
		0.5	0.3519	0.1636	0.2461	0.6024
		0.75	0.3679	**0.1677**	0.2518	0.6112
		1.0	**0.3875**	0.1623	**0.2659**	**0.6274**
	Noise	0.25	0.2941	0.1384	0.2020	0.5826
		0.5	0.2736	0.1268	0.1855	0.5624
		0.75	0.2447	0.0965	0.1602	0.5270
		1.0	0.0748	−0.0065	0.0570	0.1683
Llama 3.2 3B	Baseline	Base	0.3173	0.1672	0.2017	0.5962
		0.0	0.3270	0.1631	0.2074	0.5917
	Concepts	0.25	0.3405	0.1658	0.2238	0.6071
		0.5	0.3507	0.1707	0.2340	0.6208
		0.75	0.3649	0.1781	0.2481	0.6337
		1.0	**0.3936**	**0.1844**	**0.2733**	**0.6482**
	Noise	0.25	0.2910	0.1471	0.1721	0.5886
		0.5	0.2663	0.1404	0.1500	0.5732
		0.75	0.2366	0.1154	0.1266	0.5278
		1.0	0.0606	−0.0006	0.0589	0.1784
Llama 3.1 8B	Baseline	Base	0.3585	0.1547	0.2381	0.6011
		0.0	0.3818	0.1568	0.2482	0.5967
	Concepts	0.25	0.3963	0.1638	0.2595	0.6114
		0.5	0.4073	0.1680	0.2657	0.6247
		0.75	0.4193	**0.1705**	0.2774	0.6339
		1.0	**0.4362**	0.1623	**0.3008**	**0.6462**
	Noise	0.25	0.3339	0.1462	0.1932	0.5755
		0.5	0.3099	0.1300	0.1790	0.5538
		0.75	0.2796	0.1049	0.1418	0.5285
		1.0	0.0811	−0.0111	0.0700	0.1631

Table 3: Performance on human semantic similarity across model sizes, training weights, and concept types for models trained on **C4**. We pool final-layer token embeddings to obtain fixed-length representations, compute pairwise cosine similarity, and report Spearman correlation with human-annotated semantic similarity scores. ↑/↓ denotes higher/lower is better, and **bold** indicates the best performance for each model.

Model	Post-train Method	λ	Clustering Score ↑	ID PPL ↓	OOD PPL ↓	ID Acc ↑	OOD Acc ↑
Llama 3.2 1B	Baseline	Base	0.4530	40.61	38.56	0.3061	0.3011
		0.0	0.4502	40.49	38.71	0.3057	0.3005
	Concepts	0.25	0.4763	36.48	34.84	0.3224	0.3164
		0.5	0.5003	34.38	32.65	0.3322	0.3278
		0.75	0.5269	33.20	31.28	0.3388	0.3379
		1.0	**0.5648**	**33.11**	**30.47**	**0.3419**	**0.3441**
	Noise	0.25	0.2609	67.82	62.75	0.2639	0.2651
		0.5	0.2097	139.55	130.08	0.2159	0.2238
		0.75	0.1545	401.52	374.05	0.1405	0.1522
		1.0	0.0046	19670.63	28643.16	0.0025	0.0023
Llama 3.2 3B	Baseline	Base	0.5251	30.10	28.43	0.3391	0.3347
		0.0	0.5104	30.91	28.39	0.3372	0.3341
	Concepts	0.25	0.5367	27.93	25.62	0.3539	0.3514
		0.5	0.5564	26.46	24.12	0.3648	0.3626
		0.75	0.5733	**25.70**	23.22	0.3708	0.3723
		1.0	**0.5755**	25.78	**22.80**	**0.3717**	**0.3763**
	Noise	0.25	0.3158	52.71	47.92	0.3022	0.3022
		0.5	0.2589	108.50	99.13	0.2589	0.2632
		0.75	0.1962	299.11	264.21	0.1771	0.1920
		1.0	0.0070	19518.80	28898.47	0.0028	0.0023
Llama 3.1 8B	Baseline	Base	0.6258	25.10	22.26	0.3601	0.3640
		0.0	0.6215	25.27	22.42	0.3586	0.3626
	Concepts	0.25	0.6345	22.95	20.26	0.3750	0.3801
		0.5	**0.6395**	21.88	19.19	0.3839	0.3924
		0.75	0.6376	**21.50**	**18.68**	**0.3911**	0.4010
		1.0	0.6115	22.03	**18.68**	0.3902	**0.4011**
	Noise	0.25	0.4321	43.65	39.49	0.3286	0.3337
		0.5	0.3539	85.58	78.40	0.2950	0.3042
		0.75	0.2652	223.01	203.15	0.2217	0.2368
		1.0	0.0095	23849.59	35501.71	0.0024	0.0022

Table 4: Clustering score and content word NTP performance across model sizes, training weights, and concept types for models trained on **C4**. In-distribution (ID) refers to evaluations on held-out C4 samples, while out-of-distribution (OOD) refers to evaluations on OpenWebText. Perplexity and accuracy are measured on only content words. ↑/↓ denotes higher/lower is better, and **bold** indicates the best performance for each model.

Model	Post-train Method	λ	MEN ↑	SimLex ↑	WordSim353 ↑	STS-B ↑
Llama 3.2 1B	Baseline	Base	0.3196	0.1534	0.2211	0.5790
		0.0	0.3182	0.1540	0.2199	0.5759
	Concepts	0.25	0.3365	0.1644	0.2371	0.5891
		0.5	0.3527	**0.1699**	0.2509	0.6003
		0.75	0.3659	0.1698	0.2557	0.6099
		1.0	**0.3823**	0.1614	**0.2693**	**0.6238**
	Noise	0.25	0.2969	0.1403	0.1990	0.5878
		0.5	0.2866	0.1309	0.1913	0.5644
		0.75	0.2589	0.1091	0.1641	0.5308
		1.0	0.0753	−0.0093	0.0781	0.1697
Llama 3.2 3B	Baseline	Base	0.3173	0.1672	0.2017	0.5962
		0.0	0.3214	0.1576	0.2158	0.5934
	Concepts	0.25	0.3372	0.1669	0.2279	0.6059
		0.5	0.3483	0.1726	0.2365	0.6185
		0.75	0.3605	0.1768	0.2462	0.6286
		1.0	**0.3817**	**0.1804**	**0.2696**	**0.6435**
	Noise	0.25	0.2908	0.1441	0.1690	0.5932
		0.5	0.2771	0.1422	0.1600	0.5791
		0.75	0.2489	0.1247	0.1336	0.5265
		1.0	0.0634	−0.0071	0.0431	0.1738
Llama 3.1 8B	Baseline	Base	0.3585	0.1547	0.2381	0.6005
		0.0	0.3830	0.1634	0.2566	0.6028
	Concepts	0.25	0.3987	0.1702	0.2712	0.6160
		0.5	0.4086	**0.1768**	0.2842	0.6267
		0.75	0.4154	0.1767	0.2906	0.6368
		1.0	**0.4275**	0.1685	**0.3071**	**0.6505**
	Noise	0.25	0.3419	0.1515	0.2249	0.5872
		0.5	0.3225	0.1377	0.2017	0.5672
		0.75	0.2846	0.1159	0.1605	0.5249
		1.0	0.0909	−0.0099	0.0709	0.1923

Table 5: Performance on human semantic similarity across model sizes, training weights, and concept types for models trained on **OpenWebText**. We pool final-layer token embeddings to obtain fixed-length representations, compute pairwise cosine similarity, and report Spearman correlation with human-annotated semantic similarity scores. ↑/↓ denotes higher/lower is better, and **bold** indicates the best performance for each model.

Model	Post-train Method	λ	Clustering Score ↑	ID PPL ↓	OOD PPL ↓	ID Acc ↑	OOD Acc ↑
Llama 3.2 1B	Baseline	Base	0.4474	38.56	40.61	0.3011	0.3061
		0.0	0.4451	38.05	41.54	0.3024	0.3024
	Concepts	0.25	0.4712	33.94	37.46	0.3200	0.3183
		0.5	0.4942	31.85	35.13	0.3313	0.3275
		0.75	0.5172	30.65	33.67	0.3388	0.3358
		1.0	**0.5516**	**30.30**	**32.91**	**0.3428**	**0.3408**
	Noise	0.25	0.2712	63.97	66.20	0.2675	0.2673
		0.5	0.2147	132.84	137.04	0.2246	0.2188
		0.75	0.1566	377.89	399.66	0.1558	0.1452
		1.0	0.0033	18020.53	23964.93	0.0047	0.0039
Llama 3.2 3B	Baseline	Base	0.5247	28.43	31.00	0.3347	0.3391
		0.0	0.4997	28.03	31.53	0.3343	0.3362
	Concepts	0.25	0.5230	25.09	28.57	0.3535	0.3513
		0.5	0.5402	23.63	26.90	0.3658	0.3617
		0.75	**0.5528**	22.87	25.84	0.3741	0.3690
		1.0	0.5482	**22.77**	**25.08**	**0.3764**	**0.3734**
	Noise	0.25	0.3194	48.69	49.70	0.3047	0.3046
		0.5	0.2580	100.46	100.20	0.2663	0.2595
		0.75	0.1916	275.96	273.07	0.1933	0.1902
		1.0	0.0054	18376.56	22826.87	0.0068	0.0055
Llama 3.1 8B	Baseline	Base	0.6200	22.26	25.10	0.3640	0.3601
		0.0	0.6078	22.05	25.80	0.3644	0.3552
	Concepts	0.25	0.6203	19.84	23.45	0.3831	0.3710
		0.5	**0.6222**	18.89	22.21	0.3945	0.3829
		0.75	0.6168	**18.53**	21.51	**0.4013**	0.3912
		1.0	0.5832	18.92	**21.10**	0.3969	**0.3938**
	Noise	0.25	0.4253	39.22	41.40	0.3362	0.3302
		0.5	0.3482	77.73	79.74	0.3067	0.2967
		0.75	0.2619	202.21	217.33	0.2445	0.2293
		1.0	0.0076	22424.33	29168.22	0.0049	0.0037

Table 6: Clustering score and content word NTP performance across model sizes, training weights, and concept types for models trained on **OpenWebText**. In-distribution (ID) refers to evaluations on held-out OpenWebText samples, while out-of-distribution (OOD) refers to evaluations on C4. Perplexity and accuracy are measured on only content words. ↑/↓ denotes higher/lower is better, and **bold** indicates the best performance for each model.