Learning Concepts, Not Tokens: Self-Supervised Semantic Alignment for Language Models

Anonymous ACL submission

Abstract

The next-token prediction (NTP) objective trains language models to predict a single token at each step, even though many continuations can express the same meaning. For example, in the sentence "this sticker can be *placed* here", *positioned*, *attached*, or *put* are all plausible alternatives. While standard NTP training treats these alternatives as mutually exclusive targets, we explore a self-supervised framework that encourages models to predict concepts, approximated as sets of semantically equivalent tokens. Models trained with this concept supervision align better with human similarity judgments, improve classification, clustering, and reranking performance, and achieve comparable or stronger downstream reasoning. These gains come with lower perplexity on semantically meaningful words (Section 3.2) and only minimal increases in global perplexity, suggesting that concepts enhance semantic alignment while preserving language modeling quality.[1]

1 Introduction

Large language models (LLMs) can generate coherent text, answer questions, and perform complex reasoning, yet most are still trained with a simple objective: next token prediction (NTP). While effective in practice, NTP emphasizes surface-level token patterns, treating different wordings as distinct even when they express the same meaning (Michaelov et al., 2025).

In contrast, humans reason in terms of concepts that can be expressed through different words and contexts. For example, *dad* and *father* refer to the same entity, while words like *dog* and *pet* can evoke the same concept in context. As a result, NTP may penalize valid alternatives despite their semantic equivalence, diverging from human conceptual processing (Bachmann and Nagarajan, 2024; Liu et al., 2026b; 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-tuned models, trained on data with short contexts, show **stronger alignment with human semantic intuition**, improved performance on short-context classification, clustering, and reranking, and exhibit comparable or stronger downstream reasoning abilities. Moreover, despite not being trained to predict exact tokens, concept models achieve lower perplexity on content words (defined in Section 3.2) than all baselines, and slightly higher global perplexity over all tokens.

2 Related Work

NTP's reliance on exact token matching can 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., 2026a). Bender and Koller (2020) argue that this surface-level focus limits semantic understanding, a claim supported by failures of NTP and teacher forcing on tasks requiring entailment, planning, or multiple valid continuations (Bachmann and Nagarajan, 2024; Merrill et al., 2024). Together, these findings suggest that fluent generation does not necessarily imply semantic understanding.

Recent work has explored alternatives to standard *pretraining*, including multi-token prediction (DeepSeek-AI et al., 2025; Ahn et al., 2025; Chen et al., 2025), plan-then-generate frameworks (Mahajan et al., 2026; Wei et al., 2025), latent reasoning (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). Other approaches instead address NTP's limitations dur-

[1] Our code is available at https://anonymous.4open.science/r/learning-concepts-9025.

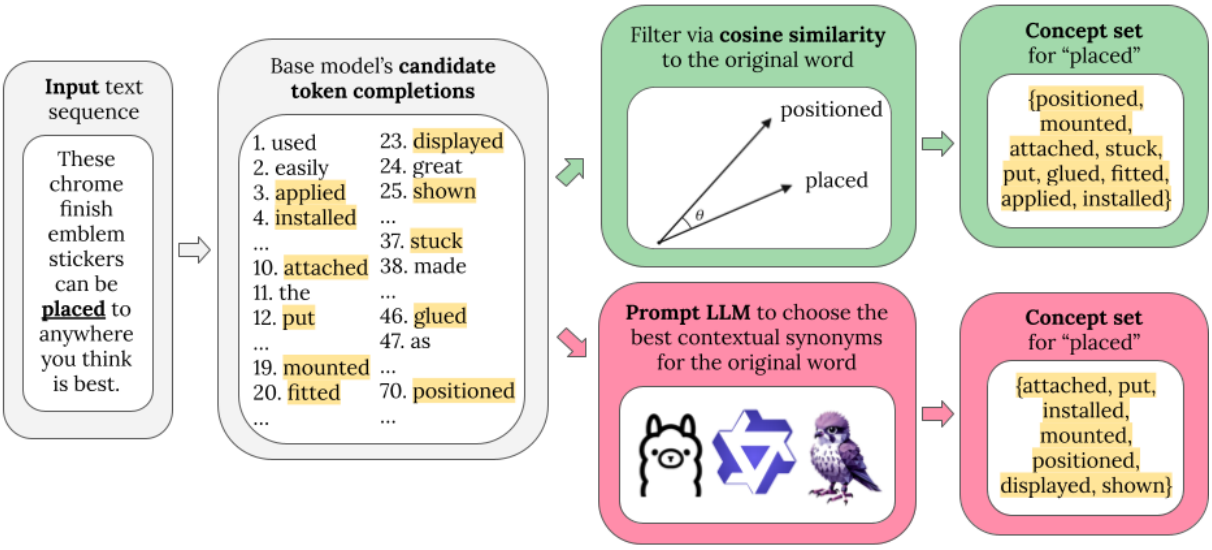

Figure 1: Our process for concept dataset generation, illustrated on one word for this sequence. We pass each sequence into a decoder model, and for each noun, verb, or adjective token that comprises an entire word, we extract the top 100 NTP completions. Then, we select the best contextual synonyms for those words using two different methods. In method 1, we compare the cosine similarities between the original word's embedding and embedding of each candidate token, and we keep the candidate tokens that have a cosine similarity of at least 0.75. In method 2, we prompt the model's corresponding instruct-tuned model to choose the best synonyms from the candidate tokens. These synonyms form the concept set for the original word in the sequence.

ing *post-training* through RLHF, DPO, instruction tuning, contrastive learning, and multitask finetuning (Ouyang et al., 2022; Wu et al., 2023; Rafailov et al., 2023). For example, semantic augmentations improve generalization (Yan et al., 2024), while lexical supervision improves word sense disambiguation (Levine et al., 2020).

The most closely related work incorporates concept-level information into the training signal through next-synonym and hypernym prediction (Iyer et al., 2026). Although effective, these methods rely on external resources such as WordNet to define concept sets. In contrast, our method derives concept structure directly from the model's embeddings and extends beyond nouns to include verbs and adjectives.

3 Training Methods

We first list the base models used (Section 3.1). We then describe both the process of enriching standard data with concept signal (Section 3.2; Figure 1), and modifying standard NTP objective function to be concept-aware (Section 3.3; Figure 2).

3.1 Base Models

Throughout the paper, we explore 3 model families and 3 model sizes within each family. We use: 1) Llama 3.2 1B, Llama 3.2 3B, Llama 3.1 8B (Grattafiori et al., 2024), 2) Qwen 3 1.7B, Qwen 3 4B, Qwen 3 8B (Team, 2025), and 3) Falcon 3 1B, Falcon 3 3B, Falcon 3 7B (Team, 2024).

3.2 Concept Dataset Creation

We use two standard datasets: AllenAI's C4 (Raffel et al., 2020) and OpenWebText (Gokaslan and Cohen, 2019), an open-source replication of WebText. From each corpus, we sample 10,000 English text sequences and split them into training, validation, and test sets using an 80/10/10 ratio. We follow Sentence Transformers (Reimers and Gurevych, 2019) and truncate sequences to 256 tokens. Models are thus trained on $10{,}000 \times 0.8 \times 256 = 2{,}048{,}000$ tokens from either C4 or OpenWebText.

We use two methods to generate concept datasets: one operates within each model, and the other uses its instruction-tuned variant. Since both are model-dependent, we generate a separate dataset for each base model in Section 3.1.

Both methods first identify token positions for concept-level supervision. We target tokens that (i) correspond to complete words rather than subwords and (ii) are nouns, verbs, or adjectives, ensuring that meaningful semantic alternatives exist.

To identify complete words, we retain tokens that begin with a space and are followed by either another space-prefixed token or punctuation,

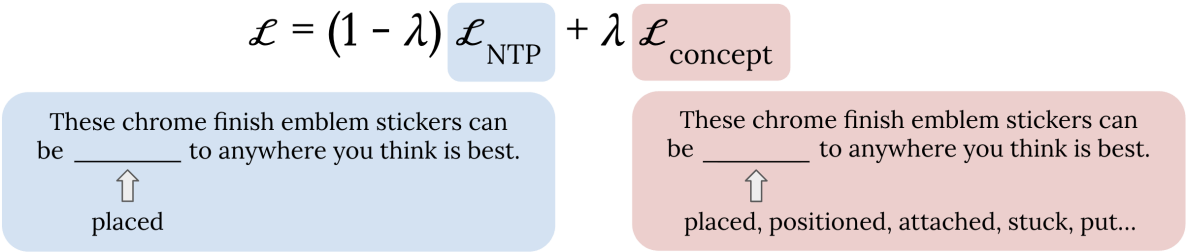

Figure 2: Our loss function incorporating both standard NTP and a concept term, balanced via a concept weight hyperparameter λ. Setting $\lambda = 0$ is pure NTP training, and increasing λ makes the models more concept-oriented.

selecting standalone words rather than subword fragments. This excludes the first word in each sequence, since lacking preceding context prevents reliable identification of contextual synonyms.

We use spaCy POS tagging and verify alignment with the tokenizer. On average, complete-word single-token nouns, verbs, and adjectives (*content words*) comprise 28% of tokens per sequence.

For each content-word position, we generate the top 100 next-token predictions as candidates that express the same concept. Our two methods then differ in how they filter this pool into a final set of contextually interchangeable tokens (Figure 1).

Method 1: Embeddings. We compute the cosine similarity between each candidate's contextual embedding and that of the original token. We discard candidates whose spaCy part-of-speech tag does not match the original. From the remaining pool, we retain candidates with cosine similarity at least 0.75, capped at 10 per content word. The resulting set serves as a discrete approximation of the concept associated with that token in its context.

Method 2: Prompting. We prompt the model's corresponding instruction-tuned variant to select which of the 100 candidates can replace the original token without changing the sentence's meaning (see Appendix A for the prompt used). We note that while Llama and Falcon provide instruction-tuned models at each of our decoder sizes, Qwen does not release instruction-tuned variants at 1.7B or 8B. We therefore use Qwen 4B Instruct across all three Qwen model sizes.

3.3 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 (e.g., "this sticker can be placed"), and let T be the original content token ("placed"). We define T^* as the set of contextual synonyms for T, including T itself: {*placed, attached, put, installed, ...*}. Instead of optimizing the likelihood of predicting the exact token T, the concept objective rewards predicting *any* token belonging to the conceptual equivalence class T^*:

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

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.

3.4 Combined Training Loss

We apply standard NTP loss to non-content words, and for content words combine NTP and concept loss via a weighted linear interpolation (Figure 2):

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

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 (see implementation details in Appendix B).

3.5 Models

We post-train our nine base models using concept loss weights $\lambda \in \{0.25, 0.5, 0.75, 1.0\}$ (training details in Appendix B). We train separately on C4 and OpenWebText, and separately on concept datasets produced by each of our two generation methods, yielding 144 total combinations (9 base models $\times$ 4 concept loss weights $\times$ 2 datasets $\times$ 2 concept dataset generation methods).

3.6 Baselines

We compare against four baselines, each designed to isolate a specific potential source of gains.

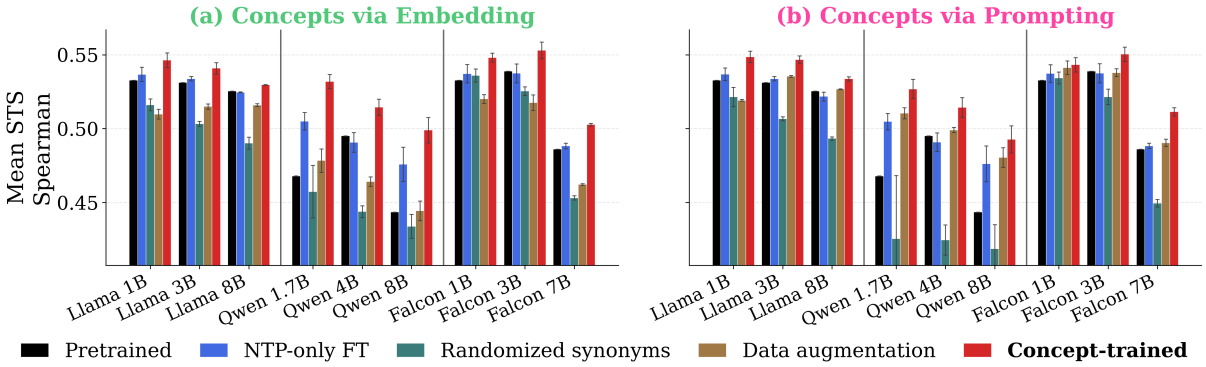

Figure 3: **Concept-trained models align more closely with human semantic intuition.** Results are averaged over the C4 and OpenWebText datasets, with error bars showing the standard error between these two. Concept models are shown at $\lambda = 1$. Randomized-synonym models are shown at $\lambda = 0.25$ and data-augmentation models at 1 epoch, which are the best-performing settings for each baseline.

Base model. The base model with no post-training, to measure the effect of post-training.

NTP. We post-train models with only NTP loss ($\lambda = 0$) on the same data, to measure the contribution of concept signal.

Randomized synonym sets. To verify that the observed gains stem from semantically coherent synonym sets rather than from lexical variation acting merely as a regularizer, we train models in which each true synonym set is replaced with a random set of the same size, sampled from the global pool of training-set synonyms. We use $\lambda \in 0.25, 0.5, 0.75, 1.0$, since the $\lambda = 0$ condition is already covered by the previous baseline.

Data-augmented NTP. To verify that gains stem from the concept objective itself rather than from increased exposure to semantically similar words, we train NTP models on a concept-augmented dataset. For each original sequence, we generate four additional variants in which every content word is replaced by a sample from its corresponding concept set. When a concept set contains at least four synonyms, samples are drawn without replacement; otherwise, sampling is performed with replacement (including the original word). We train two variants: one matched to our main models in total training steps (controlling for compute), and another matched in epochs (controlling for unique data exposure), the latter requiring five times as many training steps. This data-augmented baseline also mirrors the best-performing configuration from Iyer et al. (2026), allowing a direct comparison against the strongest prior approach.

4 Experiments & Results

We evaluate our models on four fronts: alignment with human semantic similarity judgments (Section 4.1), general-purpose embedding tasks (Section 4.2), downstream multiple-choice tasks (Section 4.3), and NTP perplexity (Section 4.4). To align our evaluation with the context length used for post-training, we restrict the evaluation to short-context tasks and defer longer-context training and evaluation to future work (see Section 6).

4.1 Increased Semantic Understanding

We evaluate alignment with human semantic judgments using the test splits of 9 Semantic Textual Similarity (STS) tasks from the Massive Text Embedding Benchmark (MTEB (Enevoldsen et al., 2025; Muennighoff et al., 2022); see implementation details in Appendix C).

Figure 3 depicts the STS results, showing that **concept-trained models achieve higher Spearman correlation with human semantic similarity judgments than all baselines**. For Qwen 4B and Falcon 3B, the NTP-only baseline shows reduced semantic alignment relative to the pretrained checkpoint, suggesting that continued token-level optimization may erode semantic structure in some architectures. In contrast, concept supervision mitigates this degradation, enabling models to recover the pretrained model's semantic alignment.

Spearman correlation increases monotonically with concept weight λ, indicating that **stronger concept supervision yields representations that are increasingly aligned with human semantic judgments** (Appendix D). In contrast, the randomized-synonym baseline peaks at $\lambda = 0.25$

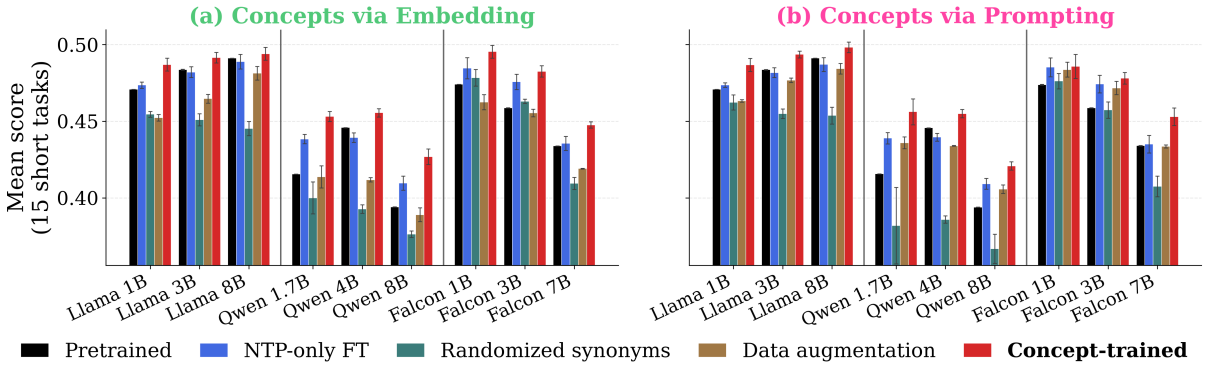

Figure 4: **Concept-trained models outperform all baselines when averaged across short-context MTEB tasks spanning classification, clustering, pair-classification, and reranking.** Scores are averaged over all 15 of these tasks and over models trained on C4 and OpenWebText; error bars show the standard error across the two training datasets. Concept models are shown at $\lambda = 1$, randomized-synonym models at $\lambda = 0.25$, and data-augmentation models at 1 epoch, which we chose as the best models for those categories based on STS performance in Figure 3.

and deteriorates as λ increases, suggesting that while a small amount of noise may act as regularization, excessive noise ultimately harms semantic coherence (Appendix D). This contrast supports our claim that the gains come from semantically meaningful concept sets, not from adding noise.

Larger Qwen and Falcon models achieve lower Spearman correlation than their smaller counterparts, consistent with Shani et al. (2026), who found that scaling does not correlate with human semantic judgments. One possible explanation is methodological: we extract embeddings by mean-pooling the final layer, whereas larger models may encode semantic information more strongly in intermediate layers. Nevertheless, we use final-layer mean pooling consistently across all models, following prior work on decoder embeddings (Muennighoff, 2022; BehnamGhader et al., 2024; Muennighoff et al., 2025a), to ensure comparability.

4.2 Generalization Across Embedding Tasks

We evaluate whether concept training improves the quality of learned *embeddings* using all MTEB (eng, v2) tasks whose inputs fit within 256 tokens, matching our short-context training regime (Section 3.2). This results in 15 tasks spanning 7 classification, 3 pair-classification, 1 reranking, and 4 clustering benchmarks across domains, including product reviews, social media, technical Q&A, and scientific abstracts. All evaluations operate directly on frozen last-layer mean-pooled embeddings through the MTEB library, which applies a trained linear probe for classification, cosine similarity for pair-classification and reranking, and unsupervised k-means for clustering.

We find that **concept training improves short-context representation over every baseline**, consistent across classification, pair-classification, reranking, and 3 of the 4 clustering tasks (Figure 4; per-category results in Appendix E). The sole exception is ArXivHierarchicalClusteringS2S (Enevoldsen et al., 2025; Muennighoff et al., 2022), where concept models score roughly two V-measure points below the baseline. Notably, this is the only hierarchical clustering benchmark in our suite, making it qualitatively distinct from the remaining tasks and potentially less aligned with the local semantic abstractions encouraged by concept training. Nevertheless, the overall pattern across the other 14 tasks remains strongly positive. As with STS, larger Qwen and Falcon models obtain lower absolute scores, likely for the same embedding-extraction reasons discussed in Section 4.1, though concept models still consistently outperform their corresponding baselines.

Taken together, these results indicate that **concept supervision yields better general-purpose representations**. The gains hold whether a task is solved by a trained probe, a cosine-similarity threshold, or unsupervised clustering, suggesting the improvement lies in the geometry of the embeddings themselves.

4.3 Improved Downstream Reasoning

In line with our short-context training regime, we evaluate our models on six multiple-choice downstream benchmarks using the Language Model Evaluation Harness (Gao et al., 2024): ARC-Challenge, ARC-Easy, OpenBookQA, HellaSwag, PIQA, and WinoGrande (Clark et al., 2018; Mi-

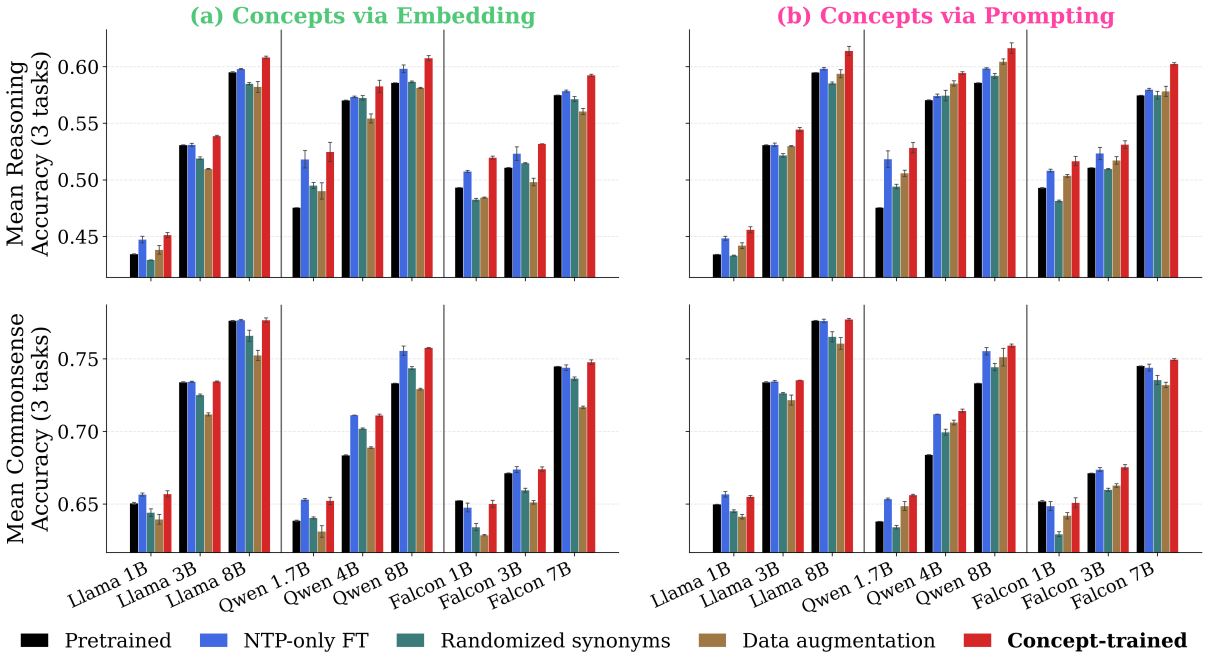

Figure 5: **Top**: Concept-trained models perform better on knowledge reasoning tasks (ARC-Challenge, ARC-Easy, OpenBookQA) compared to all baselines. **Bottom**: Concept-trained models match accuracy on commonsense tasks (HellaSwag, PIQA, WinoGrande) compared to the best-performing baseline models. Results are averaged over the C4 and OpenWebText datasets, with error bars showing the standard error between these two. Concept models are shown at $\lambda = 1$, and randomized-synonym models at $\lambda = 0.25$, the best-performing weight for that baseline.

haylov et al., 2018; Zellers et al., 2019; Bisk et al., 2020; Sakaguchi et al., 2021). We group these benchmarks into two categories: *knowledge reasoning* (ARC-Challenge, ARC-Easy, and OpenBookQA), which emphasize scientific and factual reasoning, and *commonsense reasoning* (HellaSwag, PIQA, and WinoGrande), which probe physical, linguistic, and situational commonsense.

The top part of Figure 5 shows length-normalized accuracy averaged across the *knowledge reasoning* tasks. As expected, performance improves with model size across all methods. Moreover, **concept models consistently outperform all baselines on knowledge reasoning benchmarks**, supporting the hypothesis that the gains arise from the concept objective itself rather than from alternative factors controlled for in the baselines. Higher concept weights generally lead to better performance on these benchmarks; we include results comparing reasoning performance against concept weight in Appendix F.

The bottom of Figure 5 shows accuracy averaged across the commonsense reasoning tasks.[2] Here we find no benefit from concept supervision: **concept models perform on par with the pure NTP baselines**. We hypothesize that this asymmetry in knowledge versus commonsense reasoning reflects a difference in what the two task categories demand. Knowledge reasoning relies on explicit factual and lexical content, where grouping content words with their semantic neighbors plausibly sharpens the representations a model draws on. Commonsense reasoning instead relies on implicit physical and situational expectations that are not well captured by content-word synonymy, leaving little for the concept objective to exploit.

4.4 NTP Perplexity Analysis

We analyze NTP perplexity on both in-domain (ID) held-out samples and out-of-distribution (OOD) samples from the alternate corpus (C4 or OpenWebText). Alongside standard perplexity, we report *content word perplexity*, computed only over content-word tokens. This metric isolates performance on semantically meaningful items (approximated as complete-word single-token nouns, verbs, and adjectives), excluding the function words and punctuation that are typically easier to predict:

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

[2]We report length-normalized accuracy where available; for WinoGrande, this is equivalent to raw accuracy.

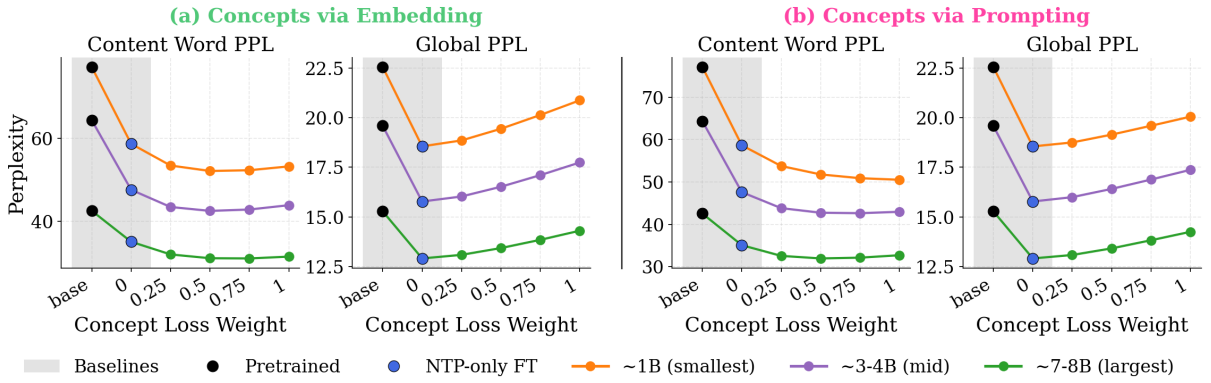

Figure 6: Concept-trained models achieve substantially lower content word perplexity than both the pretrained and NTP-only baselines, while global perplexity degrades compared to the NTP-only baseline but improves compared to the pretrained model. Results are averaged across the C4 and OpenWebText datasets and across the three model families, grouped by model size. We omit error bars because absolute perplexity values differ substantially across families due to differences in their base models; per-family plots are reported in Appendix H.

where $\mathcal{T}$ is the set of content-word tokens and $\mathrm{NLL}(t)$ is the negative log-likelihood of token t.

Figure 6 shows results aggregated over models and datasets for ID data; OOD results follow the same trends (Appendix G). **Content word perplexity decreases monotonically with concept weight across both ID and OOD data and both concept extraction methods.** Thus, concept supervision improves not only downstream behavior but also NTP at content word positions, despite training to spread probability across many plausible continuations. Additionally, content word perplexity is substantially higher than global perplexity across all conditions, indicating that content words are the harder tokens to predict in each sequence. The improvements from concept supervision are concentrated on these semantically loaded positions, where our models outperform both the pretrained models and NTP post-trained baselines.

For global perplexity, concept models generally outperform the pretrained baseline but trail the NTP baseline by ≈1-2 points. We view this as a tradeoff: **concept supervision shifts capacity toward semantically informative tokens at a modest cost on function words and other positions that contribute less to sentence meaning.**

Appendix H depicts per-family plots, which show the same trends for content-word and global ID perplexity. Qwen's base-model perplexities are substantially higher than Llama's and Falcon's, and Llama and Falcon show slight degradation on global OOD perplexity relative to the NTP baseline, small compared to the content-word gains.

5 Ablations

We perform two ablations, testing whether concept supervision drives the gains or the increased weight assigned to content words during training (Section 5.1), and studying how the amount of concept supervision affects STS Spearman correlation (Section 5.2).

5.1 Increased Concept-Word Weight via NTP

To test whether gains come from the concept objective or simply emphasizing content words, we train pure-NTP models that have a higher loss weight on content words, sweeping weights over $\{2, 5, 10, 20, 50, 100\}$ while keeping other tokens at weight 1.

For an unbiased comparison controlling for language modeling quality, Figure 7 matches each concept model ($\lambda = 1$) with the weighted-NTP model closest to it in global ID perplexity, controlling for general predictive quality. This yields a consistent choice of content-word weight 5 across all families and sizes, rather than tuning per instance.

Under this matched comparison, weighted-NTP improves content-word perplexity but worsens STS correlation. This is expected: it directly optimizes each content token, while the concept objective distributes probability over synonyms and cannot match NTP by design. However, this sharper probability does not improve semantic representations, as concept models align better with human similarity judgments. This indicates that **gains come from the concept objective, not merely emphasizing content word positions**.

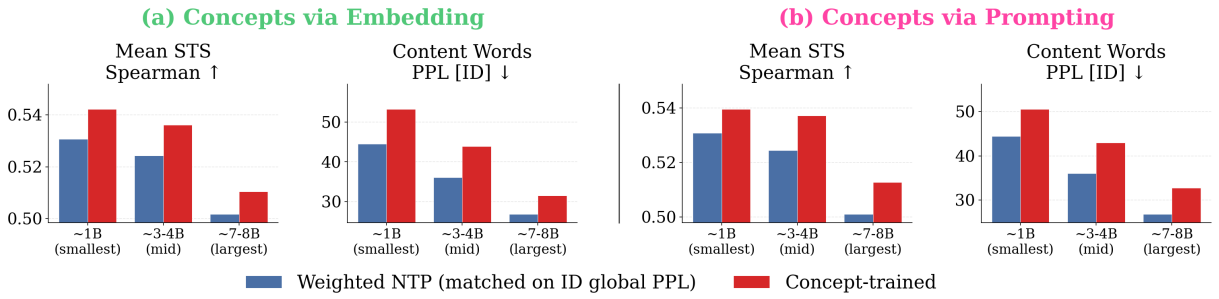

Figure 7: Concept-trained models outperform weighted-NTP models on STS Spearman correlation while weighted-NTP models achieve lower content word perplexity. Each weighted-NTP model is selected to match its corresponding concept model on in-domain global perplexity, isolating the effect of the training objective. Results are averaged over datasets and model families (see individual plots in Appendix I).

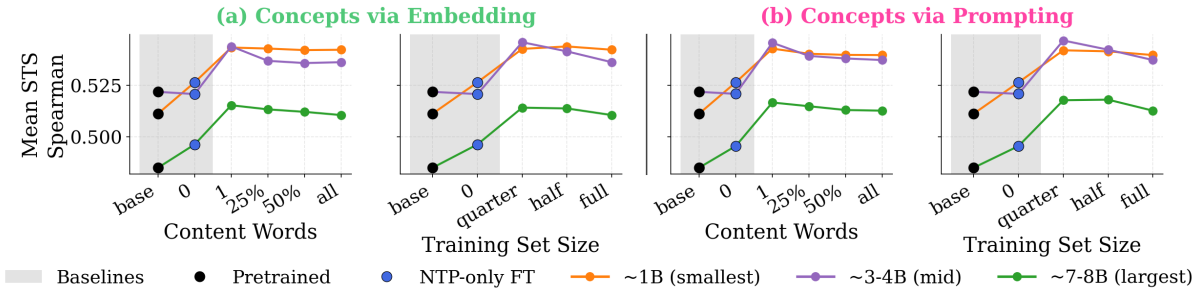

Figure 8: Concept training saturates at small data amounts on STS along both axes we vary. For the training set size plots, "0" denotes training on the full dataset at $\lambda = 0$. All models other than the pretrained and pure-NTP baselines use $\lambda = 1$. In these plots, we aggregate over datasets and model families (see individual plots in Appendix K).

5.2 Saturation at Small Data Amounts

To assess the role of concept supervision, we vary it along two axes: the fraction of supervised content words per sequence (all, 50%, 25%, a single word), and the training set size (full, 50%, 25%).

Reducing the supervised content words has little effect on STS performance. Applying concept loss to a single content word per sequence yields Spearman correlation comparable to, and sometimes higher than, applying it to all content words (Figure 8). The concept signal thus saturates quickly: **one concept per sequence suffices to convey the inductive bias**, and supervising many synonym sets may introduce overfitting.

Reducing training set size also has little effect. Spearman correlation stays essentially flat as we cut the training set to 50% and 25%, indicating that concept post-training takes effect with only a small amount of data. This finding is consistent with prior work on the data-efficiency of supervised fine-tuning (Zhou et al., 2023; Muennighoff et al., 2025b; Chen et al., 2024).

Similar saturation patterns hold for content word and global perplexity (Appendix J) and for individual model families (Appendix K).

6 Conclusions and Future Directions

We introduce a self-supervised, concept-based training objective that complements standard next-token prediction. Across three model families and three sizes, concept-trained models align more closely with human semantic similarity judgments, improve on short-context classification, clustering, and reranking tasks, strengthen knowledge reasoning, and reduce perplexity on semantically informative tokens. Comparisons against pretrained, pure-NTP, randomized-synonym, and data-augmented NTP baselines indicate that these gains stem from the concept objective itself rather than confounding factors. Notably, a small increase in global perplexity buys substantial gains on semantically meaningful tokens and downstream performance, suggesting that semantically targeted metrics can be more informative than aggregate perplexity alone.

Future work could incorporate concept supervision during pretraining, extend it to longer contexts, or apply it to multi-token expressions and multilingual data. More broadly, we view concept-level objectives as a step toward LLMs whose representations more faithfully reflect the conceptual organization underlying human language.

Limitations

Although we show promising results with concept training, our study has several limitations. First, 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 different types of tokens within sequences could yield stronger effects.

Second, our methods for constructing concept sets rely on either the decoder model's embedding structures or LLM-generated synonym clusters. While scalable, these approaches may not perfectly reflect human semantic organization, and alternative sources, such as human annotations, could provide more reliable or linguistically grounded concept groupings.

Third, our experiments focus exclusively on decoder-based models with no instruction finetuning, which limits our ability to evaluate performance under instruction prompting for downstream tasks. As a result, our evaluation tasks in this work may not capture broader ways in which concept training could be applied to improve language modeling ability.

Additionally, because we train on data truncated to 256 tokens, our models are not well-suited for long-context tasks such as retrieval. Future work can explore concept training for longer contexts.

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

Eneko Agirre, Daniel Matthew Cer, Mona T. Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *International Workshop on Semantic Evaluation*.

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: a pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, page 385–393, USA. Association for Computational Linguistics.

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 *International Conference on Machine Learning*, pages 2296–2318. PMLR.

Anil Bandhakavi, Nirmalie Wiratunga, Deepak P, and Stewart Massie. 2014. Generating a word-emotion lexicon from #emotional tweets. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 12–21, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

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*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*.

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.

Ergun Biçici. 2015. RTM-DCU: Predicting semantic similarity with referential translation machines. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 56–63, Denver, Colorado. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo 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)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. AlpaGasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*.

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*.

Xi Chen, Ali Zeynali, Chico Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw Grabowicz, Scott Hale, David Jurgens, and Mattia Samory. 2022. SemEval-2022 task 8: Multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1094–1106, Seattle, United States. Association for Computational Linguistics.

cjadams, Daniel Borkan, inversion, Jeffrey Sorensen, Lucas Dixon, Lucy Vasserman, and nithum. 2019. Jigsaw unintended bias in toxicity classification.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *Preprint*, arXiv:1803.05457.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, and 67 others. 2025. MMTEB: Massive multilingual text embedding benchmark. *arXiv preprint arXiv:2502.13595*.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2023. MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4277–4302, Toronto, Canada. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Gregor Geigle, Nils Reimers, Andreas Rücklé, and Iryna Gurevych. 2021. TWEAC: Transformer with extendable qa agent classifiers. *arXiv preprint*, abs/2104.07081.

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*.

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.

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*.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on*

Empirical Methods in Natural Language Processing, pages 1224–1234, Copenhagen, Denmark. Association for Computational Linguistics.

Ken Lang. 1995. NewsWeeder: Learning to filter netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 331–339. Morgan Kaufmann, San Francisco (CA).

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.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics.

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.

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

Yuhang Liu, Dong Gong, Yichao Cai, Erdun Gao, Zhen Zhang, Biwei Huang, Mingming Gong, Anton van den Hengel, and Javen Qinfeng Shi. 2026b. I predict therefore I am: Is next token prediction enough to learn human-interpretable concepts from data? In *The Fourteenth International Conference on Learning Representations*.

Wei Chen Maggie, Phil Culliton. 2020. Tweet sentiment extraction.

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*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Philip May. 2021. Machine translated multilingual STS benchmark dataset.

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.

James A Michaelov, Reeka Estacio, Zhien Zhang, and Ben Bergen. 2025. Not quite Sherlock Holmes: Language model predictions do not reliably differentiate impossible from improbable events. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13528–13551.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *EMNLP*.

Niklas Muennighoff. 2022. SGPT: GPT sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2025a. Generative representational instruction tuning. In *International Conference on Learning Representations*, volume 2025, pages 45544–45613.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. MTEB: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025b. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20275–20321, Suzhou, China. Association for Computational Linguistics.

James O'Neill, Polina Rozenshtein, Ryuichi Kiryo, Motoko Kubota, and Danushka Bollegala. 2021. I wish I would have loved this one, but I didn't – a multilingual dataset for counterfactual detection in product review. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7092–7108, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3982–3992.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

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 Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press.

Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. Adversarial domain adaptation for duplicate question detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1056–1063, Brussels, Belgium. Association for Computational Linguistics.

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.

Gizem Soğancıoğlu, Hakime Öztürk, and Arzucan Özgür. 2017. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33(14):i49–i58.

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*.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

TII Team. 2024. The Falcon 3 family of open models.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. TSDAE: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 671–688, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. SemEval-2015 task 1: Paraphrase and semantic similarity in Twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 1–11, Denver, Colorado. Association for Computational Linguistics.

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.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

A Prompt to Extract Concept Sets

This is the prompt fed to the instruction model to select the subset of contextual synonyms from the top 100 decoded tokens for each decoder model. 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

All models are trained using QLoRa-4bit (Dettmers et al., 2023) with a learning rate of 7×10^{-5} and batch size of 2 for 5 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. The models take about 0.5-2 hours to train on an NVIDIA A6000 GPU depending on model size and dataset.

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.

C STS Details

We use the English subset of MTEB v2. These STS tasks include STS12, STS13, STS14, STS15, STS17, STS22.v2, STSBenchmark, SICK-R, and BIOSSES (Agirre et al., 2012, 2013; Bandhakavi et al., 2014; Biçici, 2015; Cer et al., 2017; Chen et al., 2022; May, 2021; Marelli et al., 2014; Soğancıoğlu et al., 2017).

For each sequence, we obtain a representation by mean-pooling the final-layer hidden states and applying L2 normalization. MTEB's library then computes cosine similarity between sentence representations for each pair and reports Spearman correlation with the human similarity judgments. We then average the score reported using MTEB on the test splits of each of the STS datasets.

D STS by Concept Weight

We show results for STS Spearman correlation separated by concept weight, aggregated over model families in Figure 9.

We also show STS Spearman correlation results for the randomized synonym baseline models in Figure 10.

E Short-Context MTEB by Task

We report results for the 15 short-context MTEB (Enevoldsen et al., 2025; Muennighoff et al., 2022) tasks, split by category:

- **Classification (7 tasks; Figure 11):** AmazonCounterfactualClassification (O'Neill et al., 2021), Banking77Classification (Casanueva et al., 2020), MassiveIntentClassification (FitzGerald et al., 2023), MassiveScenarioClassification (FitzGerald et al., 2023), MTOPDomainClassification (Li et al., 2021), ToxicConversationsClassification (cjadams et al., 2019), TweetSentimentExtractionClassification (Maggie, 2020).

- **Pair-classification (3 tasks; Figure 12):** SprintDuplicateQuestions (Shah et al., 2018), TwitterSemEval2015 (Xu et al., 2015), TwitterURLCorpus (Lan et al., 2017).

- **Reranking (1 task; Figure 13):** AskUbuntuDupQuestions (Wang et al., 2021).

- **Clustering (4 tasks; Figure 14 for ArXivHierarchicalClusteringS2S and Figure 15 without):** ArXivHierarchicalClusteringS2S (Enevoldsen et al., 2025; Muennighoff et al., 2022), MedrxivClusteringS2S.v2 (Enevoldsen et al., 2025; Muennighoff et al., 2022), StackExchangeClustering.v2 (Geigle et al., 2021), TwentyNewsgroupsClustering.v2 (Lang, 1995).

F Reasoning Tasks by Concept Weight

We show results for the knowledge reasoning tasks separated by concept weight, aggregated over model families in Figure 16.

We also show knowledge reasoning results for the randomized synonym baseline models in Figure 17.

G Out-of-distribution NTP Perplexity

We report both ID and OOD results together for NTP perplexity, aggregated over all model families, in Figure 18.

H NTP Perplexity by Model Family

We show results for NTP perplexity over the Llama, Qwen, and Falcon families separately in Figure 19, Figure 20, and Figure 21 respectively.

I Weighted NTP Comparison by Model

We show results for weighted NTP compared with concept models for each model separately in Figure 22.

J Saturation on Perplexity

We observe saturation at small data amounts on both content word perplexity and global perplexity (Figure 23).

K STS Saturation by Model Family

We report results per model family for STS Spearman correlation as we vary the amount of content words that receive concept supervision and the amount of training data. The plots for Llama, Qwen, and Falcon are in Figures 24, 25, and 26 respectively.

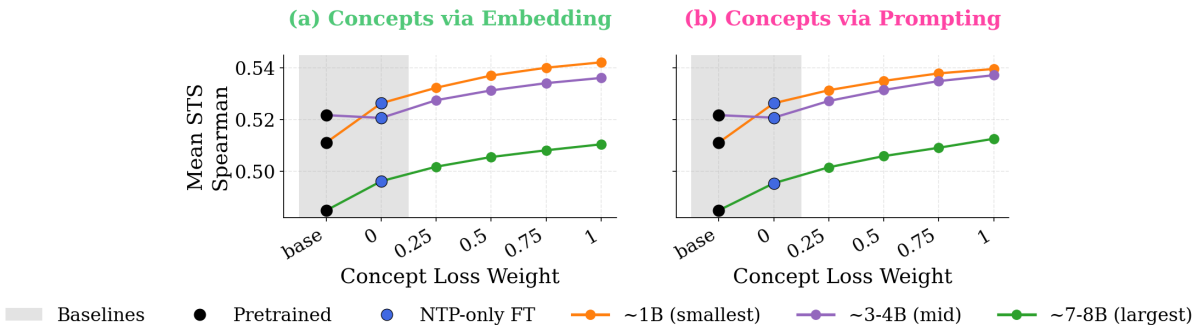

Figure 9: STS Spearman correlation increases as concept weight increases, indicating that a higher degree of concept supervision allows models to align more strongly with human semantic intuition.

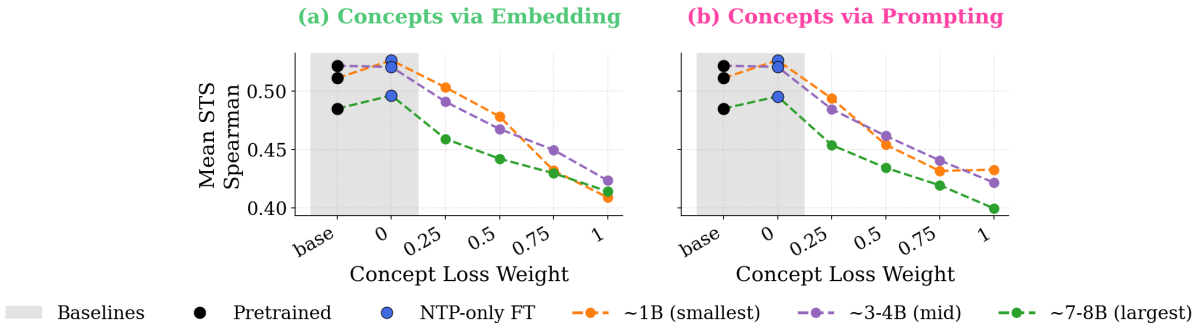

Figure 10: STS Spearman correlation decreases as concept weight increases for the models with randomized synonym sets. Thus, without semantic groupings of concept sets, model representations become more damaged as we increase the weight on these noisy concept sets.

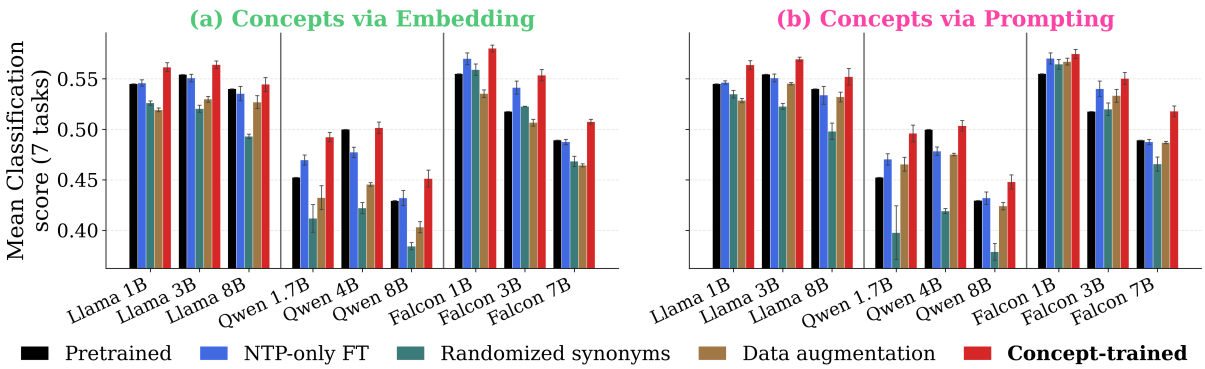

Figure 11: Concept-trained models outperform all baselines on classification tasks. Scores are averaged over 7 classification tasks and over models trained on C4 and OpenWebText; error bars show the standard error across the two training datasets. Concept models are shown at $\lambda = 1$, randomized-synonym models at $\lambda = 0.25$, and data-augmentation models at 1 epoch, which we chose as the best models for those categories based on STS performance in Figure 3.

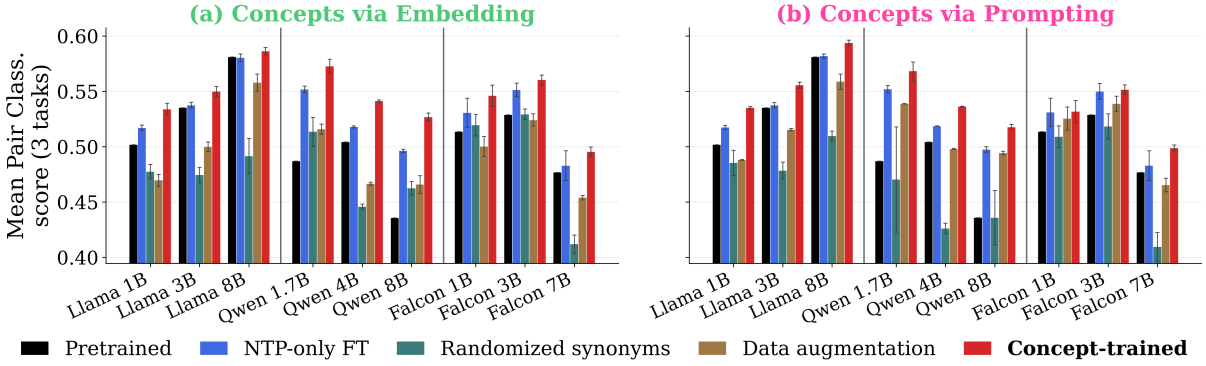

Figure 12: Concept-trained models outperform all baselines on pair-classification tasks. Scores are averaged over 3 pair-classification tasks and over models trained on C4 and OpenWebText; error bars show the standard error across the two training datasets. Concept models are shown at $\lambda = 1$, randomized-synonym models at $\lambda = 0.25$, and data-augmentation models at 1 epoch, which we chose as the best models for those categories based on STS performance in Figure 3.

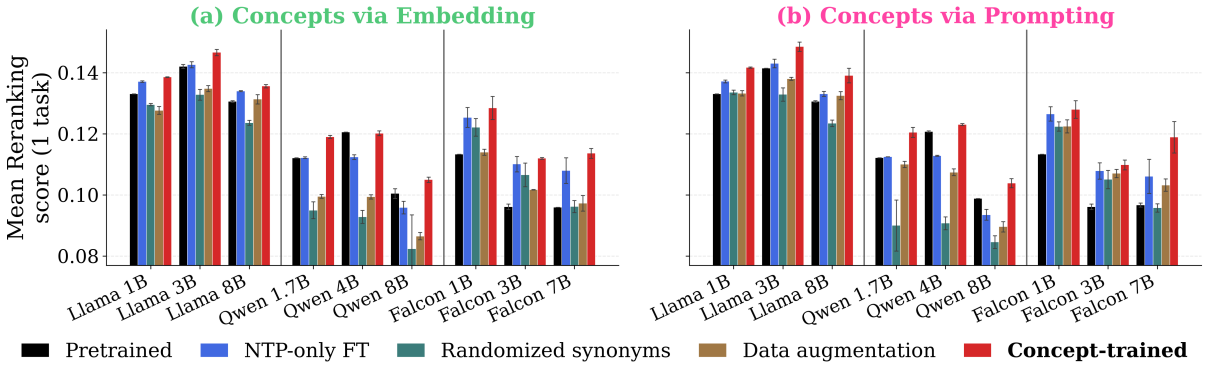

Figure 13: Concept-trained models outperform all baselines on the reranking task. Scores are averaged over models trained on C4 and OpenWebText; error bars show the standard error across the two training datasets. Concept models are shown at $\lambda = 1$, randomized-synonym models at $\lambda = 0.25$, and data-augmentation models at 1 epoch, which we chose as the best models for those categories based on STS performance in Figure 3.

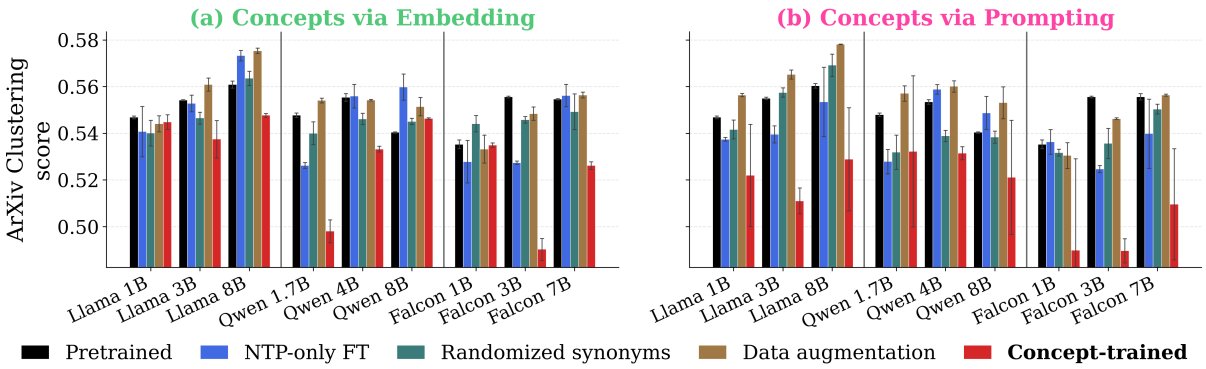

Figure 14: Concept-trained models generally trail the baselines on the arXiv clustering task. Scores are averaged over models trained on C4 and OpenWebText; error bars show the standard error across the two training datasets. We note that the error bars for this task are often large compared to the error bars in our other figures. Concept models are shown at $\lambda = 1$, randomized-synonym models at $\lambda = 0.25$, and data-augmentation models at 1 epoch, which we chose as the best models for those categories based on STS performance in Figure 3.

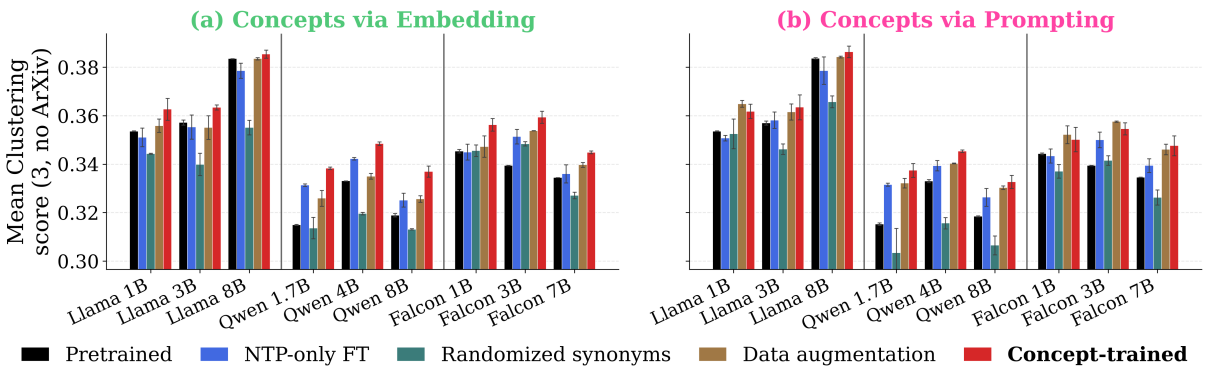

Figure 15: Concept-trained models generally outperform baselines on the remaining 3 clustering tasks, with small exceptions for Llama 1B, Falcon 1B, and Falcon 3B on the prompting side. Scores are averaged over these 3 tasks and over models trained on C4 and OpenWebText; error bars show the standard error across the two training datasets. Concept models are shown at $\lambda = 1$, randomized-synonym models at $\lambda = 0.25$, and data-augmentation models at 1 epoch, which we chose as the best models for those categories based on STS performance in Figure 3.

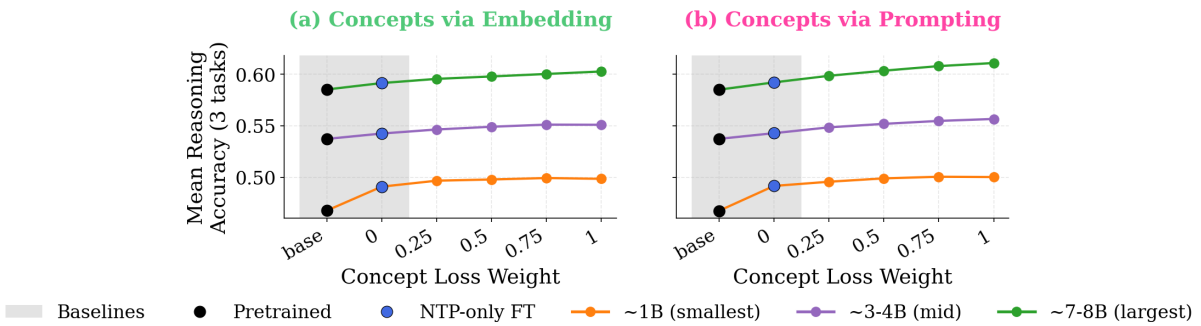

Figure 16: Reasoning accuracy increases as concept weight increases, indicating that a higher degree of concept supervision allows models to learn better factual reasoning connections. We note that the relative gains are small and that there is slight degradation at $\lambda = 1$ compared to $\lambda = 0.75$ for some models; nonetheless, general trends hold of concept training improving performance over both the pretrained and $\lambda = 0$ baselines.

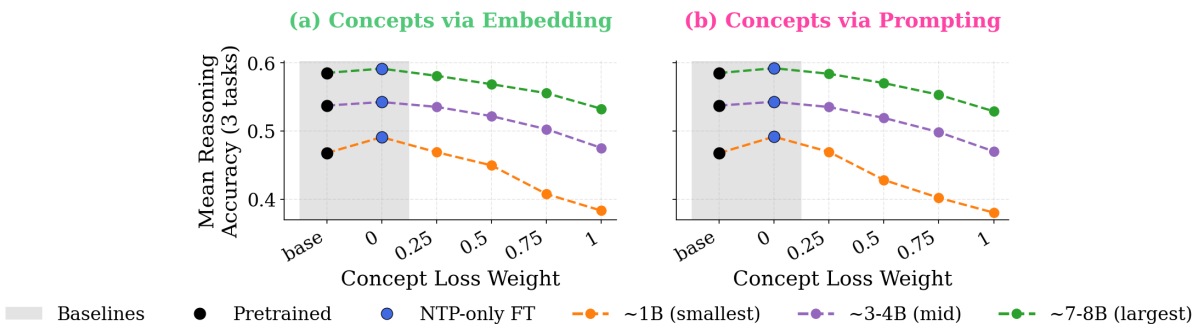

Figure 17: Reasoning accuracy decreases as concept weight increases for the models with randomized synonym sets. Thus, without semantic groupings of concept sets, model representations become more damaged as we increase the weight on these noisy concept sets.

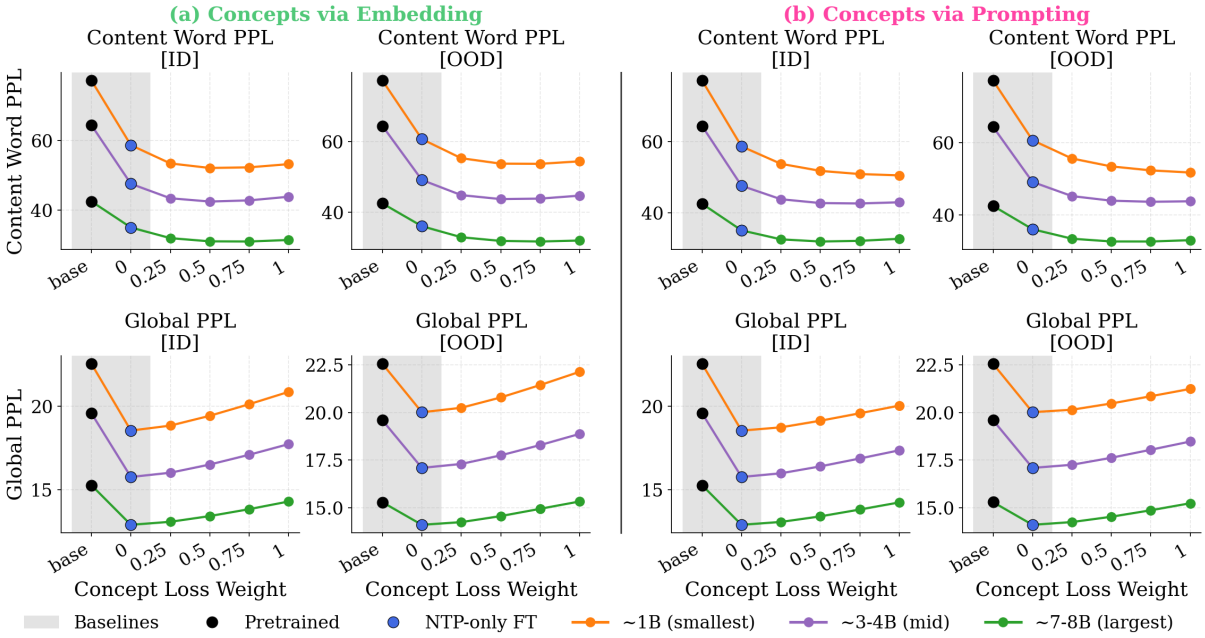

Figure 18: Concept-trained models achieve substantially lower content word perplexity than both the pretrained and NTP-only baselines both ID and OOD, while global perplexity degrades compared to the NTP-only baseline but improves compared to the pretrained model. Results are averaged across the C4 and OpenWebText datasets and across the three model families, grouped by model size. We omit error bars because absolute perplexity values differ substantially across families due to differences in their base models.

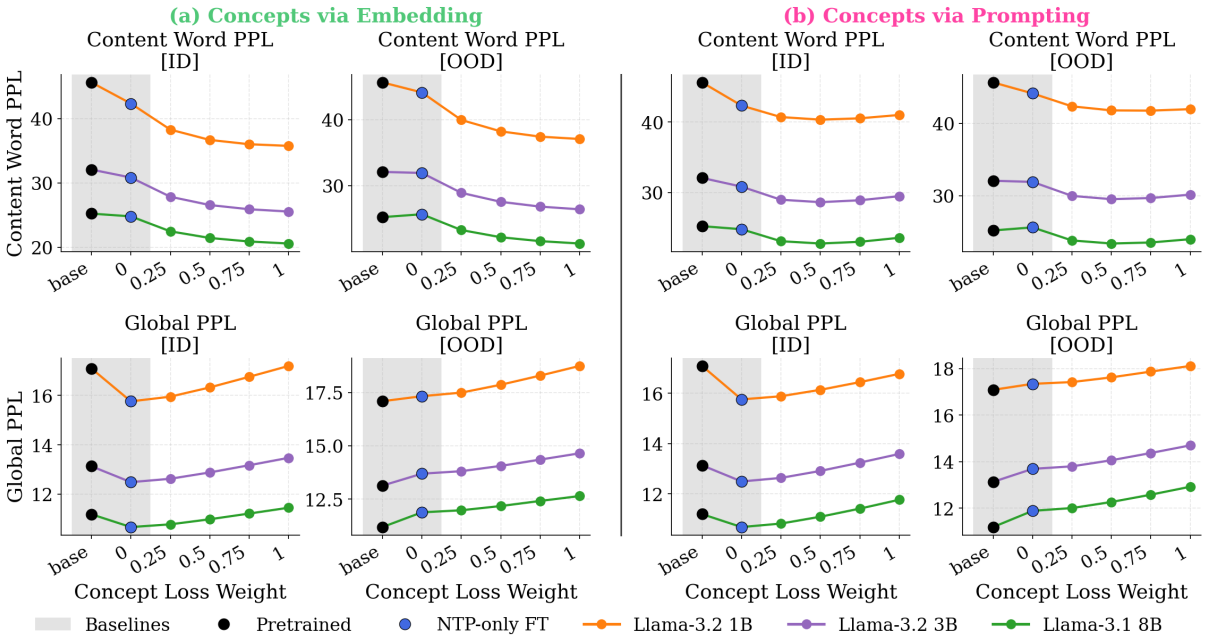

Figure 19: These plots show content word perplexity and global perplexity results for the Llama family. Content word perplexity improves both ID and OOD compared to both baselines. Global ID perplexity at higher concept weights is comparable to the pretrained baseline but 1-2 points higher compared to the NTP-only baseline. Global OOD perplexity monotonically increases as concept weight increases, but the rise is small compared to the drop in content word perplexity. The prompting models appear to plateau for content word perplexity earlier than the embedding models.

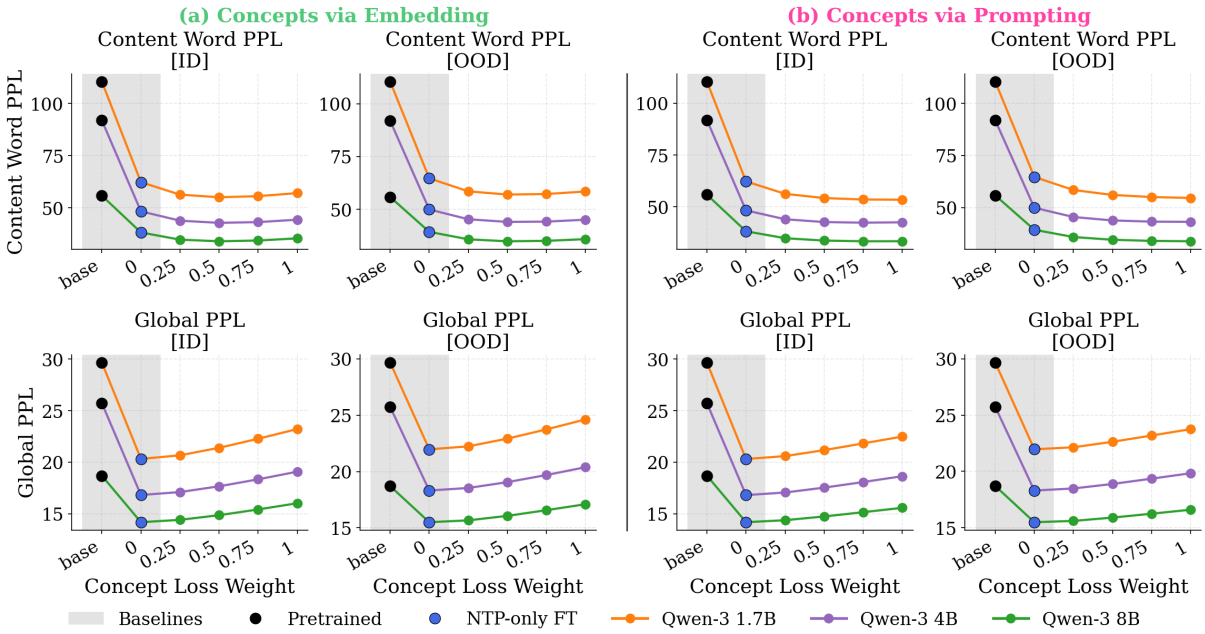

Figure 20: These plots show content word perplexity and global perplexity results for the Qwen family. Qwen's pretrained model has substantially higher content word perplexity and global perplexity both ID and OOD compared to the NTP-only baseline. As concept weight increases, content word perplexity decreases monotonically for the prompting models but plateaus around $\lambda = 0.5$ for the embedding models. Concept-trained models all outperform the pretrained baseline on global perplexity both ID and OOD. These observations are notably different compared to the Llama models, as the Llama models' global OOD perplexity degraded compared to the pretrained models (both with concept supervision and without).

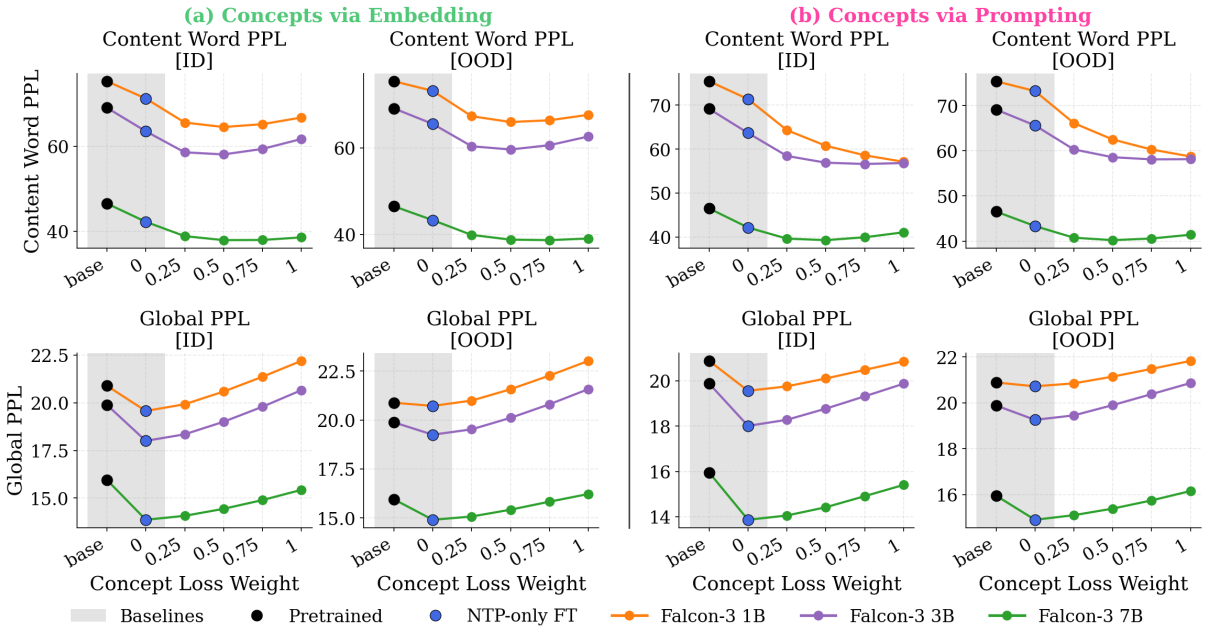

Figure 21: These plots show content word perplexity and global perplexity results for the Falcon family. Similar to the Qwen models, the embedding models for Falcon also plateau and start degrading around concept weight $\lambda = 0.5$ while some of the prompting models continue to improve as concept supervision increases. The global perplexity behaves more similarly to the Llama models, where the global ID perplexity degrades compared to the NTP baseline but is comparable to the pretrained baseline. Concept-trained models are either comparable to the pretrained baseline or 1-2 points higher on global OOD perplexity.

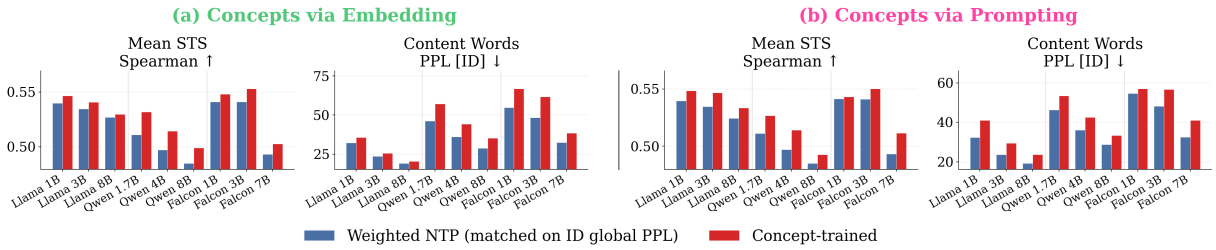

Figure 22: For every model, we see that the STS Spearman correlation for concept-trained models outperforms that of the weighted-NTP models. Conversely, the content word perplexity of the weighted-NTP models is always better than our concept-trained models, which makes sense because the weighted-NTP models specifically emphasize next token prediction at content word positions.

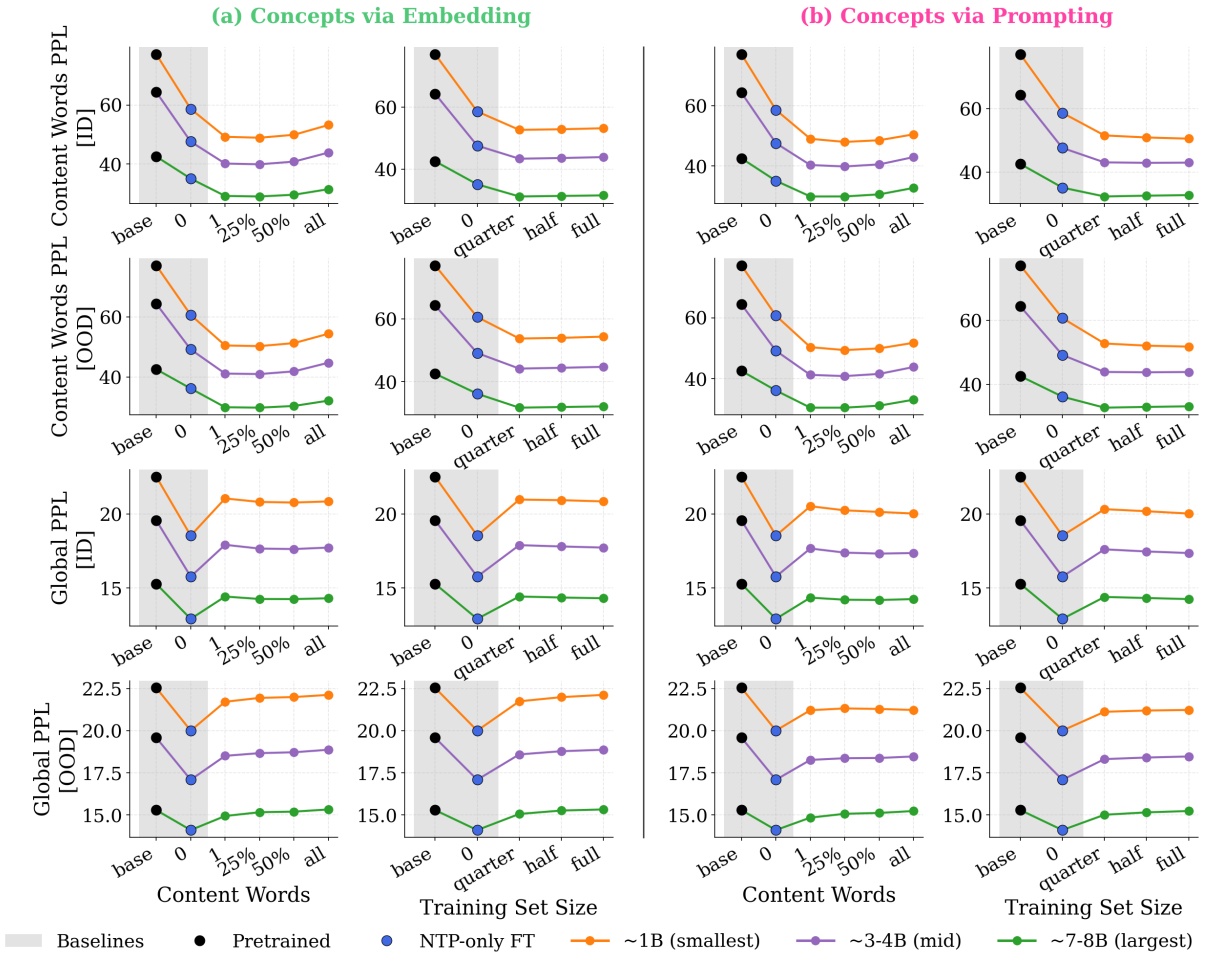

Figure 23: These plots show content word perplexity and global perplexity performance at various amounts of concept supervision. Content word ID perplexity saturates and begins overfitting when using concept supervision on only 25% of content words, while content word OOD perplexity stays relatively constant past this point. For global perplexity, we observe that the pure NTP-trained model achieves the best performance. As the amount of concept supervision increases, the global perplexity stays relatively constant both ID and OOD for all model sizes. Results are averaged over models trained on the C4 and OpenWebText datasets, as well as aggregated over model sizes in different families.

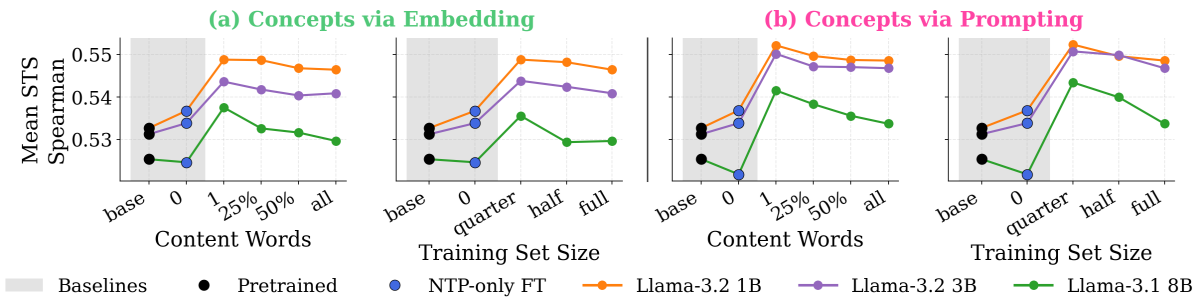

Figure 24: For Llama models, STS Spearman correlation saturates and begins overfitting after only receiving concept supervision on 1 content word per sequence, indicating that concept supervision is strong enough to affect alignment with human semantic judgments with only a sparse signal. Similarity, STS Spearman correlation also achieves its highest value with only 2,000 samples in the training set (quarter) as opposed to larger training set sizes (half, full).

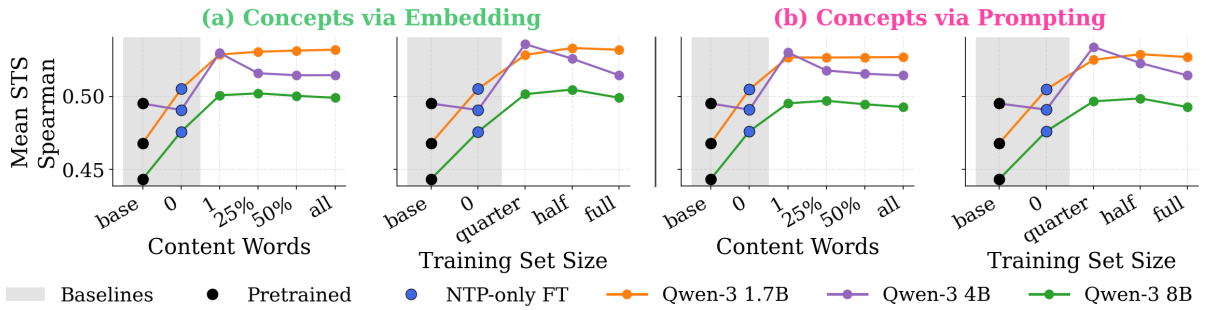

Figure 25: STS Spearman correlation for the Qwen models generally begins to plateau after receiving concept supervision on 1 content word per sequence and a quarter of the entire training set size. Contrary to Llama, Qwen does not overfit as severely.

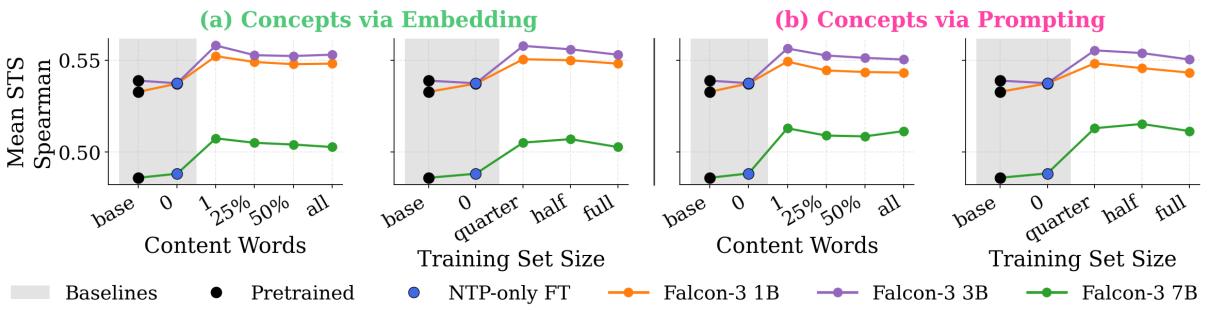

Figure 26: STS Spearman correlation for the Falcon models also achieves its highest values at 1 content word per sequence receiving concept supervision and a quarter of the training set size. At higher degrees of concept supervision, the performance slightly degrades, indicating overfitting.