

# Sentence-Level Foresight in Small Language Models: A Diagnostic Study of Learning Signals and Calibration

Ahmet Rıfat Öztürk

Yağız Ekrem Dalar

Nedim Mutlu Sezer

Feyzi Arda Salihoğlu

Ömer Faruk Aksoy

## Abstract

Small language models provide a useful setting for separating *signal acquisition* from *behavioral usefulness*. We ask whether a DistilGPT-2-scale autoregressive language model can learn a sentence-level representation that favors a plausible next sentence over self-generated loop-like alternatives, and whether that representation independently improves generation. ForesightLM-v2 combines ordinary causal language modeling with a contrastive next-sentence embedding objective and a margin loss over self-loop hard negatives. We specify the full preprocessing, future-head architecture, hard-negative construction, scoring functions, thresholds, and decoding protocol. Across three WikiText-103 seeds and three WritingPrompts seeds, the held-out positive–negative cosine gap is consistently positive. However, generation reveals a calibration gap. On a fixed five-candidate pool, removing the learned future term changes only 2 of 120 selections, while generic, random, and shuffled score controls remain close. A newly matched training-free Base-c5 control—using the same temperature, top- $p$ , maximum generation length, candidate count, and seed schedule—slightly outperforms future-calibrated c5 on the main automatic diagnostics. Local coefficient sensitivity also shows that conclusions near the nominal generic reranker are stable to the repetition and length weights. We therefore make a deliberately narrow claim: small autoregressive models can acquire a measurable sentence-level future signal, but the present evidence does not show that this learned signal is the main cause of generation improvements.

## 1 Introduction

Small language models are useful not only as compressed systems, but also as scientific instruments. Their failures are frequent, inexpensive to reproduce, and easier to audit across training, generation, and control conditions. This makes them well

suited to a question that can be obscured in larger systems: when an auxiliary representation is measurable during training, does it actually cause better behavior at generation time?

We study this question through sentence-level future prediction. Standard autoregressive language modeling optimizes next-token likelihood, yet many weak-model failures are visible at a coarser scale: a continuation may remain locally fluent while repeating a semantic state, drifting from the prompt, or failing to progress. We define a *future signal* operationally as a context-conditioned vector whose cosine similarity is higher for an appropriate next-sentence representation than for a loop-like model-generated alternative. This definition is diagnostic rather than cognitive: we do not claim that semantic similarity itself is undesirable, nor that all repetition is incoherent. Repetition can be natural when a writer clarifies, emphasizes, or elaborates. Our hard negatives instead target high-repetition model outputs as a tractable proxy for weak-model degeneration.

ForesightLM-v2 augments a DistilGPT-2 checkpoint with a sentence-level projection head. At a context boundary, the last-token hidden state from the final transformer layer is projected into the 384-dimensional space of a frozen all-MiniLM-L6-v2 sentence encoder. A batch-contrastive loss encourages the projected state to identify the actual following sentence, and an additional margin loss separates positive futures from self-loop hard negatives mined from model generations.

The central contribution is not a new decoding algorithm or a state-of-the-art coherence result. Instead, we separate five questions that are often conflated: (1) can the future signal be learned, (2) does it emerge during short training, (3) does raw auxiliary training change generation, (4) does a learned future score improve candidate selection beyond generic reranking, and (5) do the conclusions sur-

vive random, shuffled, no-future, and training-free controls? This framing is consistent with the BabyLM workshop’s broader interest in learning dynamics, data-efficient modeling, and analysis of small systems (Choshen et al., 2026). This is a non-competition workshop study rather than a BabyLM Challenge leaderboard submission.

Our main findings are mixed but informative. Training-side hard-negative gaps are positive across three WikiText-103 seeds and three WritingPrompts seeds. Generation-side results are much less specific: generic five-candidate reranking accounts for most of the semantic/progression improvement, the learned future term changes only 2/120 selections, and random or shuffled future-score controls remain close. A matched Base-c5 control confirms that this conclusion is not an artifact of mismatched sampling settings. We refer to this mismatch between representational learning and generation-time influence as a *calibration gap*.

### Contributions.

- We give a fully specified sentence-level future-prediction objective for a small causal LM, including context boundary handling, projection architecture, contrastive loss, hard-negative margin loss, and all major training hyperparameters.
- We define and document the self-loop mining pipeline and explicitly separate gold-positive from pseudo-positive hard-negative pairs.
- We show positive held-out future/hard-negative gaps across three WikiText-103 seeds and three WritingPrompts seeds, plus a short training-step sensitivity analysis.
- We test causal attribution at generation time with no-future, future-only, random, shuffled, lexical, coefficient-sensitivity, and matched training-free Base-c5 controls.
- We report the negative result directly: the learned future score is measurable, but it is weakly coupled to five-candidate selection under the present interface.

## 2 Related Work

### Small models as controlled learning systems.

TinyStories constructs a synthetic corpus restricted to child-accessible vocabulary and shows

that models with fewer than 10M parameters, including very shallow transformers, can generate multi-paragraph stories with strong fluency and consistency (Eldan and Li, 2023). Its main strategy is to simplify the data distribution so that coherent generation becomes learnable at small scale. Our goal is different: we keep a standard natural-text setting and ask whether an auxiliary sentence-level signal can be acquired by a weak causal LM, then test whether that learned representation actually affects generation. BabyLM similarly motivates small-scale language modeling as a setting for studying sample efficiency and learning behavior rather than only deployment efficiency (Choshen et al., 2026).

### Text degeneration and negative supervision.

Likelihood-trained neural generators can produce repetitive or otherwise degenerate continuations, and decoding strategy alone can substantially change those behaviors (Holtzman et al., 2020). Unlikelihood training addresses degeneration by explicitly reducing probability assigned to undesirable tokens or sequences and has been shown to reduce repetition while preserving language-modeling quality (Welleck et al., 2020). Our hard-negative objective is related in motivation but differs in level: it does not directly penalize token probabilities for a repeated phrase. Instead, it asks a sentence-level predicted future vector to separate an appropriate continuation from a loop-heavy model-generated sentence.

### Future prediction in representation space.

Contrastive Predictive Coding learns representations by predicting future latent samples using a contrastive objective with negatives (van den Oord et al., 2018). We adopt the same broad principle at sentence scale. Rather than predicting a future acoustic or visual latent, the model predicts the frozen embedding of the next sentence. Sentence-BERT provides an efficient embedding space in which semantically meaningful sentence vectors can be compared with cosine similarity (Reimers and Gurevych, 2019). This choice is computationally convenient, but it also limits the method: sentence-level semantic similarity can miss factual consistency, entity identity, discourse relations, and the distinction between legitimate re-statement and degeneration.

**Candidate selection as a confound.** Decoding and reranking can improve generation without

changing model parameters. Contrastive generation methods likewise demonstrate that candidate scoring can strongly alter output properties (Su et al., 2022). This creates a causal attribution problem for our setting: if a five-candidate selector improves generation, the gain may come from candidate diversity and a generic semantic score rather than the learned future representation. Our evaluation therefore treats generic, random, shuffled, no-future, future-only, and training-free matched reranking as central controls rather than peripheral ablations.

### 3 Method

#### 3.1 Sentence-pair construction

For WikiText-103 (Merity et al., 2017), we use the raw wikitext-103-raw-v1 text. We normalize common WikiText tokenization artifacts (e.g., @-@, @, @, and @. @) and split non-heading text using punctuation followed by whitespace and an uppercase letter, digit, quote, or apostrophe. We keep sentences containing 5–80 lexical tokens and at least 20 characters. Each valid adjacent pair forms

$$c = s_i, \quad s^+ = s_{i+1}. \quad (1)$$

We retain 30,000 training pairs and 2,000 validation pairs per domain. WritingPrompts (Fan et al., 2018) is processed analogously over story text. This simple rule is intentionally reproducible but imperfect: abbreviation-like punctuation can occasionally create spurious boundaries, which we treat as a preprocessing limitation rather than manually correcting examples.

#### 3.2 Future prediction head

Let the causal LM process the concatenation of the context and its following sentence. We separately tokenize the context to identify its boundary index. Let  $H^{(L)}$  be the final transformer-layer hidden states and  $i_c$  the last context-token index after truncation. The context representation is

$$h_c = H_{i_c}^{(L)}. \quad (2)$$

A two-layer trainable projection maps this vector to the 384-dimensional frozen sentence-encoder space:

$$\hat{z} = \text{norm}(W_2 \tanh(W_1 h_c + b_1) + b_2), \quad (3)$$

where  $\text{norm}$  denotes  $\ell_2$  normalization. The target vector is

$$z^+ = \text{norm}(E(s^+)), \quad (4)$$

with the frozen all-MiniLM-L6-v2 Sentence-Transformers encoder, kept fixed throughout training.

#### 3.3 Contrastive future loss

For a batch of  $B$  context–future pairs, we compute

$$a_{ij} = \frac{\hat{z}_i^\top z_j^+}{\tau}, \quad (5)$$

with  $\tau = 0.07$ . Because all vectors are normalized, the dot product is cosine similarity. The future loss is batch-wise cross entropy:

$$L_{\text{future}} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(a_{ii})}{\sum_{j=1}^B \exp(a_{ij})}. \quad (6)$$

The core objective is

$$L_{\text{core}} = L_{\text{LM}} + \lambda_f L_{\text{future}}, \quad (7)$$

with  $\lambda_f = 0.08$ .

#### 3.4 Self-loop hard negatives

We additionally mine loop-heavy model generations. For a generated continuation  $y$ , the mining score is

$$R_{\text{loop}}(y) = R_4(y) + R_5(y) + R_{\text{adj}}(y) + R_{\text{pair}}(y), \quad (8)$$

where  $R_4$  and  $R_5$  are repeated 4-gram and 5-gram ratios, and  $R_{\text{adj}}$  and  $R_{\text{pair}}$  are semantic repetition diagnostics already stored with the generation artifact. We retain generations with  $R_{\text{loop}} \geq 0.08$ . Within a retained generation, we first choose a sentence containing the most frequent repeated 4-gram. If no repeated 4-gram exists, we choose the sentence with the highest  $R_4 + R_5$ . This sentence becomes  $s^-$ . Thus, the negative criterion is *loop-heavy generation*, not semantic similarity alone.

The resulting pool contains 280 pairs. Of these, 146 inherit a gold next sentence from WikiText-103 or WritingPrompts generation exports. The remaining 134 originate from existing reranking exports that do not carry a gold next sentence; for those examples, the first generated sentence not selected as the hard negative is used as a pseudo-positive. We report this composition because the pool is a diagnostic supervision set rather than a clean corpus of 280 gold next-sentence triples. No synthetic fallback pairs were needed in the final pool.

For a positive  $s^+$  and hard negative  $s^-$ , the margin loss is

$$L_{\text{hn}} = \max(0, m - \cos(\hat{z}, E(s^+)) + \cos(\hat{z}, E(s^-))) \quad (9)$$

with  $m = 0.10$ . The full v2 objective is

$$L = L_{\text{LM}} + 0.08L_{\text{future}} + 0.10L_{\text{hn}}. \quad (10)$$

We do not claim these coefficients are optimal. They are fixed nominal settings from the experimental pipeline; Section 5.4 reports available sensitivity analyses rather than post-hoc tuning them for the camera-ready results.

### 3.5 Optimization

All main v2 runs initialize from DistilGPT-2 and jointly optimize the causal LM and future projection. We use AdamW with learning rate  $5 \times 10^{-5}$ , weight decay 0.01, gradient clipping at 1.0, a linear learning-rate schedule with 50 warmup steps, 1000 optimization steps, LM/future batch size 16, hard-negative batch size 32, maximum sequence length 192, and seed 7, 42, or 123. The hard-negative pool is shuffled with the run seed and split 80/20, giving 224 training and 56 held-out hard-negative examples per run. Because the split itself is seed-dependent, our multi-seed result should be read as stability across independently seeded training/held-out splits, not repeated evaluation on one identical set of 56 triples.

## 4 Generation-Time Evaluation

### 4.1 Five-candidate reranking

The future head is **not** a token-level decoder. For each of 120 WikiText validation prompts, we sample five continuations and select one after generation. Main v2 sampling uses temperature 0.9, top- $p$  0.95, maximum input length 96, maximum 120 new tokens, repetition penalty 1.0, and no no-repeat-ngram constraint. Candidate seeds follow a deterministic prompt/candidate schedule.

We embed the prompt, complete continuation, and generated sentences with the same frozen MiniLM encoder. Let the generated continuation contain sentences  $y_1, \dots, y_n$ . The generic score is

$$s_{\text{generic}}(y) = p(y) - 0.60r(y) + 0.60q(y) - 0.05\ell(y), \quad (11)$$

where

$$p(y) = \cos(E(\text{prompt}), E(y)), \quad (12)$$

$$r(y) = \frac{1}{n-1} \sum_{i=1}^{n-1} \mathbf{1}[c_i > 0.90], \quad (13)$$

$$q(y) = \frac{1}{n-1} \sum_{i=1}^{n-1} \mathbf{1}[0.35 \leq c_i \leq 0.88], \quad (14)$$

$$c_i = \cos(E(y_i), E(y_{i+1})), \quad (15)$$

$$\ell(y) = \frac{|N_{\text{words}}(y) - 105|}{105}. \quad (16)$$

For a one-sentence continuation we set the adjacent-sentence rates to zero. The progression band is therefore a heuristic for *related but non-duplicate* adjacent sentences: very low similarity suggests an abrupt semantic jump, while very high similarity suggests semantic stagnation.

The learned future feature compares the context-predicted future vector with the first generated sentence:

$$f(y) = \cos(\hat{z}(c), E(y_1)). \quad (17)$$

The future-calibrated score used in the actual main decoding job is

$$s_{\text{future}}(y) = s_{\text{generic}}(y) + 0.10f(y). \quad (18)$$

We use 0.10 as a small fixed additive coefficient and do not present it as an optimized value. Crucially, the no-future ablation removes this term entirely, so the main causal conclusion does not depend on arguing that 0.10 is optimal.

### 4.2 Diagnostics and controls

Our automatic diagnostics are defined directly from the same measurements: PSim =  $p(y)$ ; topic drift =  $1 - p(y)$ ; Prog =  $q(y)$ ; LexLoop is repeated-4-gram ratio plus repeated-5-gram ratio; semantic-loop failure is triggered when adjacent semantic repetition exceeds 0.10 or pairwise semantic repetition exceeds 0.05; and WeakProg is triggered when a continuation has at least two sentences and  $q(y) < 0.55$ . These are controlled diagnostics, not substitutes for human evaluation. Topic drift is algebraically redundant with PSim, so the main tables report PSim and omit the duplicate column.

We compare raw baseline, raw core, raw v2, generic v2 c5, a scale-matched random future-score control, future scores shuffled within a prompt, future scores shuffled across prompts, future-calibrated c5, a lexical-penalty c5 variant,

| Domain     | Seed | LM ↓   | Future ↓ | Gap ↑ | Sat. ↑ |
|------------|------|--------|----------|-------|--------|
| WikiText   | 7    | 3.9395 | 2.7228   | .0584 | .3393  |
| WikiText   | 42   | 3.9384 | 2.7180   | .0614 | .4107  |
| WikiText   | 123  | 3.9386 | 2.7347   | .0599 | .3571  |
| WritingPr. | 7    | 3.9437 | 2.8010   | .0687 | .3929  |
| WritingPr. | 42   | 3.9436 | 2.8237   | .0736 | .3929  |
| WritingPr. | 123  | 3.9428 | 2.8173   | .0606 | .3929  |

Table 1: Training-side diagnostics after 1000 steps. Gap is mean positive cosine minus hard-negative cosine. Sat. is the fraction of held-out triples whose positive-negative gap reaches the training margin  $m = 0.10$ . Each seed also induces its own 80/20 hard-negative split.

future-only selection, and a training-free matched Base-c5 control. Random scores are sampled to match the empirical mean and standard deviation of real v2 future scores; shuffling preserves the observed score distribution while breaking candidate alignment.

## 5 Results

### 5.1 The training-side signal is reproducible

Table 1 shows a consistent positive held-out margin gap across both domains. On WikiText, the three gaps lie between .0584 and .0614; on WritingPrompts they lie between .0606 and .0736. This supports a narrow representation-learning claim: under the stated objective, the context-conditioned projection learns to score positives above the mined negatives on held-out triples. It does *not* establish that the hard-negative term uniquely causes the effect, because we do not include a shuffled-hard-negative training control.

We also train WikiText seed 42 for fewer steps from the same setup. At 250, 500, and 1000 steps, the held-out gap is .0262, .0491, and .0614, respectively; the corresponding margin-satisfaction rates are .0714, .3571, and .4107. The signal is therefore detectable before the final checkpoint and strengthens during training.

### 5.2 Generation gains are mostly selection gains

Table 2 separates training effects from candidate-selection effects. Raw v2 changes the failure profile but does not consistently improve all diagnostics. In contrast, selecting among five candidates yields large improvements in prompt similarity and progression. Yet the learned future term is not the specific explanation: generic c5, ran-

dom future scores, shuffled future scores, and the aligned future score remain close.

The matched Base-c5 comparison strengthens this conclusion. An earlier Base-c5 artifact used different generation length and temperature settings, which made the comparison unnecessarily confounded. We therefore regenerated the baseline with the same main v2 sampling settings and the same candidate-seed schedule. The matched training-free baseline reaches .5080 PSim, .8765 Prog, .0083 semantic-loop rate, and .0583 weak-progression rate, slightly outperforming v2 future c5 on all five reported diagnostics. We interpret this as evidence that candidate generation plus generic reranking is a strong independent mechanism. It does not invalidate the training-side signal; it prevents attributing five-candidate gains to that signal.

### 5.3 The future term is weakly coupled to selection

On the fixed v2 five-candidate pool, removing the future term changes only 2 of 120 selections. The no-future and with-future rows have PSim.4949 vs. .4954 and Prog.8735 vs. .8727. A prompt-level paired bootstrap gives a with-future minus no-future PSim difference of 0.00047 with a 95% interval  $[-0.00009, 0.00150]$ ; for progression the difference is  $-0.00080$  with interval  $[-0.00238, 0]$ . Future-only selection raises mean future similarity from approximately .277 to .337, but lowers PSim to .4370, Prog to .6446, and raises semantic-loop and weak-progression rates to .0833 and .3583. The learned feature therefore contains information that can be maximized, but maximizing it is not equivalent to selecting the best continuation under the present diagnostics.

At candidate level, future similarity has descriptive Spearman correlations of .3569 with prompt similarity, .1548 with progression,  $-.0692$  with semantic-loop failure, .0302 with lexical-loop penalty, and .3029 with an oracle-style semantic score. Because candidates are grouped five-per-prompt, we do not treat naive 600-candidate  $p$ -values as independent-sample significance tests. A complementary prompt-level diagnostic reaches only 18.3% top-1 agreement between future-score selection and oracle semantic selection, with mean within-prompt Spearman correlation .111. These results support the same calibration-gap interpretation without relying on anti-conservative candidate-level inference.

| Method                         | PSim $\uparrow$ | Prog $\uparrow$ | LexLoop $\downarrow$ | SemLoop $\downarrow$ | WeakProg $\downarrow$ |
|--------------------------------|-----------------|-----------------|----------------------|----------------------|-----------------------|
| Baseline raw                   | .4197           | .6675           | .1130                | .0667                | .3750                 |
| Core raw                       | .4177           | .6674           | .1146                | .0917                | .3583                 |
| v2 raw                         | .4274           | .6361           | .0770                | .0667                | .4000                 |
| v2 generic c5                  | .4954           | .8727           | .1024                | .0250                | .0667                 |
| v2 random future               | .4958           | .8721           | .1018                | .0267                | .0683                 |
| v2 shuffled within prompt      | .4927           | .8768           | .0997                | .0250                | .0583                 |
| v2 shuffled across prompts     | .4949           | .8735           | .1022                | .0250                | .0667                 |
| v2 future c5                   | .4954           | .8727           | .1027                | .0250                | .0667                 |
| v2 future c5 + lexical control | .4913           | .8672           | .0546                | .0083                | .0667                 |
| Matched Base c5, no future     | <b>.5080</b>    | <b>.8765</b>    | <b>.0817</b>         | <b>.0083</b>         | <b>.0583</b>          |

Table 2: Generation diagnostics on 120 WikiText prompts. The matched Base-c5 control uses the same 5-candidate count, temperature .9, top- $p$  .95, maximum input length 96, maximum 120 new tokens, and candidate seed schedule as the main v2 decoding setup. Generic, random, shuffled, and learned-future v2 selectors remain close.

#### 5.4 Coefficient sensitivity

The generic reranker contains heuristic coefficients, so we evaluate whether the main conclusion depends on one narrow setting. Holding the progression weight at its nominal .60, varying the repetition weight from .30 to .90 and the length weight from 0 to .10 changes only 0–2 of 120 selected candidates relative to the nominal  $w_r = .60, w_q = .60, w_\ell = .05$  selector. Changing the progression weight has a larger effect:  $w_q = .30$  or .90 changes roughly 10–20 selections, as expected because progression is a major term in the selector. The important attribution result is stable in the local nominal neighborhood: repetition and length coefficients are not acting as fragile hidden switches.

We also have a training-side sensitivity over  $\lambda_h \in \{.05, .10, .20\}$  for WikiText seed 42. LM loss remains effectively unchanged (3.9385, 3.9384, 3.9383), while the held-out gap increases (.0512, .0614, .0701). We therefore use .10 as the nominal run rather than claiming it is optimal; the existence of a positive separation signal is not unique to that exact hard-negative weight. We do not have an analogous multi-value sweep for  $\lambda_f = .08$ , which remains a limitation.

#### 5.5 Concrete examples

Table 3 illustrates the hard-negative construction. The negative is not simply a semantically related next sentence; it is extracted from a visibly repetitive model continuation. The example also highlights why the signal should be interpreted as a de-generation diagnostic rather than a general theory of discourse: a human-written restatement could be legitimate even when semantically similar.

Generation examples tell the same story. For a prompt describing lobster “crusher” and “cut-

ter” claws, generic c5 and future-calibrated c5 select the same continuation containing repeated “crusher” phrases, whereas the lexical-control selector chooses a different, less lexically repetitive candidate. Conversely, several prompts show generic and future c5 choosing exactly the same candidate, and failure cases remain in which future-calibrated c5 produces entity confusion or topical drift. We therefore use qualitative examples only to make the experimental conditions concrete, not as evidence overriding the aggregate controls.

### 6 Discussion

The experiments distinguish three levels of evidence. First, the auxiliary representation is learnable: the predicted future vector scores held-out positives above mined negatives across seeds and domains. Second, the signal is semantically structured: it correlates with prompt anchoring and an oracle-style semantic score. Third, behavioral usefulness is much weaker: under five-candidate reranking, the future term changes almost no selections and does not outperform generic or matched training-free controls.

This is a useful negative result for small-model analysis. If we had reported only raw-v2 versus future-c5 generation, the improvement could easily be described as evidence that sentence-level foresight improves coherence. The random, shuffled, no-future, future-only, coefficient-sensitivity, and matched Base-c5 controls show that such wording would be too strong. The present model learns a direction in sentence-embedding space, but it does not learn a complete decoding policy. A good continuation must balance prompt relevance, semantic progression, non-repetition, length, factual consistency, and entity continuity; a single

|                     |                                                                                                                                                          |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Context             | The closest relative of <i>H. gammarus</i> is the American lobster, <i>Homarus americanus</i> .                                                          |
| Gold next sentence  | The two species are very similar, and can be crossed artificially, although hybrids are unlikely to occur in the wild since their ranges do not overlap. |
| Mined hard negative | <i>H. gammarus</i> is the American lobster, known as the American lobster, known as the American lobster, known as the American lobster, ...             |

Table 3: One WikiText hard-negative example from the final mined pool. The negative is chosen from a high-repetition model generation.

future-similarity scalar captures only part of that structure.

The results also clarify the role of self-loop negatives. We use highly repetitive generations because they are frequent weak-model failures and provide cheaply mined contrast. We do *not* assume that all repetition is bad or that every semantically similar next sentence should be pushed away. The hard-negative builder can produce false positives with respect to discourse quality, and 134 of 280 pool items use pseudo-positives rather than gold next sentences. Consequently, our strongest supported statement is about learnable loop separation under this diagnostic construction, not about a universal notion of future coherence.

## 7 Limitations

Our evaluation is automatic and covers 120 WikiText prompts; we do not report human judgments of coherence, factuality, or discourse quality. The frozen MiniLM target space can miss entity consistency, factual contradictions, and discourse relations. The sentence splitter is rule-based and occasionally treats abbreviation-like punctuation as a boundary. The hard-negative pool is small and mixed-source, and 134/280 triples use a generated pseudo-positive because their source reranking exports lack a gold next sentence. Each training seed also uses a different 80/20 hard-negative split, so seed stability combines optimization and split variation. We do not run a shuffled-hard-negative training control, a gold-only hard-negative training control, or a  $\lambda_f$  sweep; these are the most direct ways to test whether the specific auxiliary construction, rather than easier repetition artifacts, causes the training-side gap. Finally, the matched Base-c5 result shows that the learned future term is not necessary for the strongest automatic generation scores in this setup. Accordingly, we make no claim that ForesightLM-v2 solves coherence or that its future term is the principal cause of generation improvements.

## 8 Conclusion

ForesightLM-v2 demonstrates a distinction between learning a future-oriented sentence representation and using that representation behaviorally. A DistilGPT-2-scale model trained with next-sentence contrastive prediction and self-loop hard negatives learns a consistent held-out positive-negative margin across WikiText-103 and WritingPrompts seeds. Yet under a controlled five-candidate generation interface, the learned future score has little independent effect: removing it changes only 2/120 selections, random and shuffled controls remain close, and a matched training-free Base-c5 reranker slightly outperforms it on the reported automatic diagnostics. We therefore interpret sentence-level foresight as a measurable but incompletely calibrated representation. For small-model research, this gap is itself informative: acquiring a diagnostic signal is easier than converting it into reliable generation behavior.

## References

- Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Jaap Jumelet, Tal Linzen, Aaron Mueller, Suchir Salhan, Raj Sanjay Shah, Alex Warstadt, and Ethan Gotlieb Wilcox. 2026. [BabyLM turns 4: Call for papers for the 2026 babyLM workshop](#). *Preprint*, arXiv:2602.20092.
- Ronen Eldan and Yuanzhi Li. 2023. Tinstories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898. Association for Computational Linguistics.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text de-generation. In *International Conference on Learning Representations*.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture mod-

els. In *International Conference on Learning Representations*.

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*, pages 3982–3992. Association for Computational Linguistics.

Yixuan Su, Tian Lan, Yan Liu, Fangyu Liu, Dani Yogatama, Yan Wang, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. In *Advances in Neural Information Processing Systems*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

## A Reproducibility Details

| Training setting               | Value              |
|--------------------------------|--------------------|
| Base checkpoint                | DistilGPT-2        |
| Sentence encoder               | all-MiniLM-L6-v2   |
| Sentence embedding dim.        | 384                |
| Train pairs/domain             | 30,000             |
| Validation pairs/domain        | 2,000              |
| Training steps                 | 1000               |
| LM/future batch size           | 16                 |
| Hard-negative batch size       | 32                 |
| Max sequence length            | 192                |
| Learning rate                  | $5 \times 10^{-5}$ |
| AdamW weight decay             | 0.01               |
| Warmup steps                   | 50                 |
| Gradient clipping              | 1.0                |
| $\lambda_f$                    | 0.08               |
| $\lambda_h$                    | 0.10               |
| Contrastive temperature $\tau$ | 0.07               |
| Hard-negative margin $m$       | 0.10               |
| Hard-negative pool             | 280                |
| Hard-negative train/val        | 224 / 56           |

Table 4: Main training configuration.

## B Additional Sensitivity Results

**Generic-score neighborhood.** On the fixed v2 five-candidate pool, with  $w_q = .60$ , all combinations  $w_r \in \{.30, .60, .90\}$  and  $w_\ell \in \{0, .05, .10\}$  differ from the nominal selector on at most two of 120 prompts. For example,  $(w_r, w_q, w_\ell) = (.30, .60, .05)$  changes one selection and gives PSim.4981, Prog.8707, SemLoop .0333, and

| Generation setting        | Value     |
|---------------------------|-----------|
| Prompts                   | 120       |
| Candidates/prompt         | 5         |
| Temperature               | 0.9       |
| Top- $p$                  | 0.95      |
| Max input tokens          | 96        |
| Max new tokens            | 120       |
| Repetition penalty        | 1.0       |
| No-repeat n-gram size     | 0         |
| Future rerank coefficient | 0.10      |
| Generic $w_r$             | 0.60      |
| Generic $w_q$             | 0.60      |
| Generic $w_\ell$          | 0.05      |
| Length target             | 105 words |

Table 5: Main v2 five-candidate generation and reranking configuration. The matched Base-c5 control uses the same sampling values.

| $\lambda_h$ | LM ↓   | Future ↓ | Gap ↑ | Sat. ↑ |
|-------------|--------|----------|-------|--------|
| .05         | 3.9385 | 2.7214   | .0512 | .3571  |
| .10         | 3.9384 | 2.7180   | .0614 | .4107  |
| .20         | 3.9383 | 2.7148   | .0701 | .4643  |

Table 6: Hard-negative loss-weight sensitivity for Wiki-Text seed 42.

WeakProg.0667;  $(.90, .60, .05)$  changes zero selections relative to the nominal candidate IDs and therefore has the same selected outputs. Increasing  $w_q$  to .90 changes 10 selections and increases Prog to approximately .896 while lowering PSim to approximately .478; decreasing it to .30 changes 16–20 selections and has the opposite trade-off. This confirms that progression weighting matters, while the local repetition/length settings are not fragile.

## C Future-Term Ablation Details

| Selector         | PSim ↑ | Prog ↑ | Sem. ↓ | Weak ↓ | Fut. ↑ |
|------------------|--------|--------|--------|--------|--------|
| v2 c5, no future | .4949  | .8735  | .0250  | .0667  | .2751  |
| v2 c5, + future  | .4954  | .8727  | .0250  | .0667  | .2769  |
| Future only      | .4370  | .6446  | .0833  | .3583  | .3368  |

Table 7: Fixed-pool future-term ablation. The future term changes only 2/120 selections.

## D Qualitative Examples

**Generic and future selectors often agree.** For the prompt “*It is closely related to the American lobster, *H. americanus*.*”, generic c5 and future-calibrated c5 select the same candidate. The continuation remains topically related but contains factual and taxonomic noise. This example illus-

trates both candidate-selection gains over raw decoding and the lack of a unique future-term effect.

**Lexical control addresses a different failure.**

For the prompt describing “crusher” and “cutter” lobster claws, generic c5 and future c5 select a continuation that repeats variants of “*The crusher is a variation of the crusher*”. Adding the lexical penalty chooses a different candidate and lowers the measured surface-loop score. This matches the aggregate result that future similarity and lexical repetition are only weakly associated.

**Failure case.** Some future-calibrated selections remain semantically or factually confused. One taxonomy prompt yields repeated names and historically inconsistent entities despite a reasonable semantic score. We include such cases because the automatic metrics should not be mistaken for factual or human-level coherence judgments.