

Does a Data2vec Auxiliary Help Small Masked Language Models? A BabyLM-2026 Strict-Small Study

Ivan Dostál Martin Benedikt Robin Klubarski

Faculty of Information Technology

Czech Technical University in Prague

{dostaiva,benedma2,klubarob}@fit.cvut.cz

Abstract

We ask whether a low-weight data2vec-style Exponential Moving Average (EMA) self-distillation auxiliary ($\lambda=0.1$, $\tau=0.999$) improves a small masked language model at the BabyLM-2026 Strict-Small scale. Our base model is a 33M-parameter LTG-BERT encoder with GPT-BERT-style hybrid masked/causal training, trained from scratch on the official 10M-word corpus in 2.3 GPU-hours on a single NVIDIA L40S. A substantial gain in Entity-Tracking during early development from adding the auxiliary motivated the study, but disappeared in later re-evaluation. In a comparison across 6 seeds, no evaluation task changed significantly (aggregate effect $p=0.29$), and the early gain is best explained by seed noise. Several variants intended to strengthen the auxiliary do no better. We attribute the model’s competitive scores to its hybrid training recipe and to EMA weight-averaging, both already established.

1 Introduction

The BabyLM Challenge (Warstadt et al., 2023) asks how much linguistic and cognitive competence a language model can acquire from the amount of text a child would encounter. The 2026 Strict-Small track caps the corpus at 10M words and training at 10 epochs so a strong result must come from the objective, architecture, or data.

We study one candidate for such an objective: self-distillation with an Exponential Moving Average (EMA) teacher (Grill et al., 2020; Baevski et al., 2022; Assran et al., 2023), in which the model regresses onto a slowly updated running average of itself and so needs no extra text. One closely related attempt used latent bootstrapping as the main training goal at BabyLM scale and found it useful but erratic (Samuel, 2023). We test the lighter-touch version: the EMA objective as a small auxiliary ($\lambda=0.1$) on top of a GPT-BERT-trained LTG-BERT

encoder, which already combines masked and next-word prediction (Charpentier and Samuel, 2024).

The result is negative. Under controlled re-evaluation the pilot gain disappears. Toggling only the auxiliary weight ($\lambda=0.1 \leftrightarrow 0$), with recipe, seed, and the official 2026 evaluation pipeline held fixed across six matched seeds, the auxiliary changes *no* zero-shot task significantly (Table 2): the chance-anchored aggregate moves $+0.03$ ($p=0.29$) and Entity Tracking $+0.1$ ($p=0.89$). The dramatic development gap is best explained by seed variation on the pre-update evaluation pipeline (Section 4.1).

Contribution. Our contribution is a negative result with these findings:

- (i) a seed-controlled ablation showing a low-weight data2vec auxiliary does not improve a hybrid-trained LTG-BERT at 10M words on any of seven zero-shot tasks (Section 4);
- (ii) evidence that the model’s above-chance behavior comes from the recipe and from EMA weight-averaging.
- (iii) four single-axis variants meant to strengthen the auxiliary (a higher weight, a decoupled higher-rate mask, a wider target window, a slower teacher), none of which help (Section 4.2);
- (iv) a candidate mechanism (Section 4.2): the auxiliary predicts the teacher at the same masked positions used by masked prediction, so it is largely redundant with it, and a sequence-length guard switches it off for the final phase of training.

We also could not reproduce the Entity-Tracking score on the GPT-BERT-mixed baseline’s model card (39.95); our re-run of that recipe reaches

20.1,¹ so all our Entity-Tracking claims are within-pipeline comparisons against our own no-auxiliary baseline, both scored with the official pipeline (see Limitations).

2 Related Work

BabyLM. Strong Strict-Small entries modify the objective or architecture: ELC-BERT (2023 winner) learns per-layer weights (Charpentier and Samuel, 2023); GPT-BERT (2024 winner) unifies fill-in-the-blank and next-word prediction with a single shared head (Charpentier and Samuel, 2024). We adopt its recipe (but with two separate heads) and its LTG-BERT backbone (Samuel et al., 2023).

Self-distillation. In self-distillation the model learns from targets produced by a copy of itself. BYOL (Grill et al., 2020) showed that predicting the features of an EMA copy, a “teacher” that trails the “student” as a slow running average of its weights, learns useful representations without collapsing. data2vec (Baevski et al., 2022, 2023) applies the same idea to masked inputs across modalities, regressing onto an average of the teacher’s top layers; I-JEPA (Assran et al., 2023) predicts teacher features for hidden regions; LLM-JEPA (Huang et al., 2025) applies JEPA objectives to LLMs via paired views. We follow the data2vec formulation (Section 3).

Closest work. BootBERT (Samuel, 2023), our main inspiration, combined latent bootstrapping with MLM at BabyLM scale with the weighting inverted relative to ours (bootstrapping as the main objective, MLM as the auxiliary), and found it useful but erratic: better (Super)GLUE, mixed BLiMP, and worse linguistic generalization. We ask whether the reverse weighting, a small latent-prediction auxiliary on top of a strong masked-prediction recipe, behaves any better.

Known risks. Joint-embedding objectives risk representational collapse (Bardes et al., 2022), which EMA alone does not reliably prevent (Mo and Tong, 2024; Halvagal et al., 2023), and multi-objective training is sensitive to loss weighting and gradient interference (Kendall et al., 2018; Chen et al., 2018; Yu et al., 2020). The latter is consistent with the narrow safe range of λ we observe (Section 4.2).

¹This was before the July 2026 change to the Entity-Tracking evaluation dataset, and we did not train on multiple GPUs as the baseline intended.

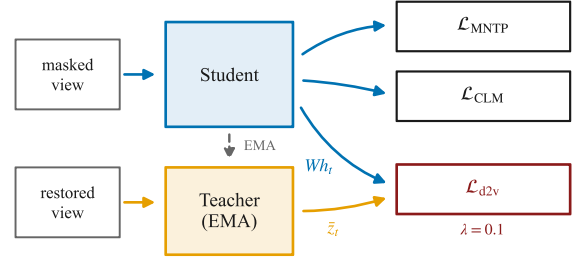


Figure 1: Training setup. MNTP and CLM cross-entropies train the student; the auxiliary $\mathcal{L}_{d2v}$ (Eq. 2) regresses Wh_t onto the teacher target $\bar{z}_t$ (mean of the layer-normalized top-8 teacher layers, stop-gradient). The teacher is an EMA copy of the student and encodes the restored view. The I-JEPA variant replaces the auxiliary branch (Appendix B).

3 Method and Setup

Data and tokenizer. We train on the official BabyLM-2026 Strict-Small corpus, with the same published 16,384-token BPE vocabulary as the GPT-BERT-mixed baseline.²

Base model. The backbone is the LTG-BERT encoder (Samuel et al., 2023) with the GPT-BERT-mixed release’s modifications: 12 layers, hidden size 384, 6 attention heads, feed-forward size 1280, inputs up to 512 tokens; 33.4M trainable parameters (36.2M total as exported, encoder-only LTGBertForMaskedLM). Training follows the GPT-BERT hybrid scheme (Charpentier and Samuel, 2024): most batches (93.75%) train masked next-token prediction (MNTP): labels are shifted by one position, so each masked token is predicted from the position preceding it. The remaining 6.25% train next-word prediction under a causal mask. Masked spans of 1–3 tokens (Joshi et al., 2020), rather than single tokens (Devlin et al., 2019), are sampled at a rate annealed from 30% to 15%. Unlike GPT-BERT’s single shared head, we use two separately parameterized prediction heads, both output-tied to the embeddings. The causal head is dropped at export.

The EMA auxiliary. The student is accompanied by a teacher: a full copy of the student, created at initialization and never trained by gradient descent. After every optimizer step each teacher parameter ϕ is pulled slightly toward its student counterpart θ ,

$$\phi \leftarrow \tau \phi + (1 - \tau) \theta, \quad \tau = 0.999 \text{ (constant)}, \quad (1)$$

²babylm-baseline-10m-gpt-bert-mixed.

so the teacher is an exponential moving average of the student’s own history (visualized in Figure 1). Its outputs serve as regression targets. On an MNTP batch the teacher encodes a restored view of the input (owing to the label shift, each masked target token is placed at its prediction position rather than the original sequence being rebuilt exactly), and the student, through a single linear head W at the masked positions M , regresses onto the mean $\bar{z}_t$ of the teacher’s layer-normalized top $K=8$ layers, i.e., the teacher’s contextual embedding of the target token. The submitted weights are the EMA teacher copy of the final checkpoint. We also trained a segment-level I-JEPA-style variant for comparison; it performed similarly to the data2vec variant, so we did not pursue it further (Appendix B).

Training loss. Each step minimizes

$$\mathcal{L} = \begin{cases} \mathcal{L}_{\text{MNTP}} + w_z \mathcal{L}_z + \lambda g \mathcal{L}_{\text{d2v}} & \text{MNTP} \\ \mathcal{L}_{\text{CLM}} + w_z \mathcal{L}_z & \text{CLM} \end{cases} \quad (2)$$

where:

- $\mathcal{L}_{\text{d2v}} := \frac{1}{|M|} \sum_{t \in M} \ell_\beta(W h_t, \text{sg}(\bar{z}_t))$,
- $\mathcal{L}_{\text{MNTP}}$ and $\mathcal{L}_{\text{CLM}}$ are cross-entropies,
- $\mathcal{L}_z$ is a LogSumExp² logit regularizer,
- ℓ_β the Huber loss ($\beta=4$),
- sg the stop-gradient,
- $\lambda=0.1$,
- and $g \in \{0, 1\}$ being the loss-dropout gate: $P(g=1)=0.5$ below sequence length 512 and $g=0$ in the 512 phase (measured, $\sim 47\%$ of sub-512-window batches carry the auxiliary gradient).

Training. Table 1 lists the hyperparameters used. It is mostly the same as the GPT-BERT-mixed baseline release’s published hyperparameters (with adjusted steps to match the strict-small cap and with the auxiliary added). One run takes ≈ 2.3 GPU-hours on one NVIDIA L40S. The submitted model, including all 19 intermediate checkpoints (chck_1M–chck_100M), is available on the HuggingFace Hub.³

³<https://huggingface.co/poopoobabylm/babylm-2026-strict-small-d2v-s3>

Evaluation and scoring. During development we evaluated every saved model zero-shot, without fine-tuning, on BLiMP and its supplement (Warstadt et al., 2020, 2023), EWoK (Ivanova et al., 2025), Entity Tracking (Kim and Schuster, 2023), COMPS (Misra et al., 2023), and reading-time prediction (SPR/ET) (de Varda et al., 2024). Entity Tracking, the skill we target, tests whether a model can follow the changing state of objects through a passage; each item is a five-way choice, so its chance level is 20.0, whereas the other minimal-pair tasks are binary (chance 50.0) and the reading-time scores are ΔR^2 over a baseline regression, $\times 100$ (chance 0). We report raw per-task scores plus an aggregate defined as

$$\text{agg} = \frac{1}{T} \sum_{i=1}^T \frac{s_i - c_i}{b_i - c_i}, \quad (3)$$

where s_i is our model’s score on test i , c_i its chance level, and b_i the score of the official GPT-2 anchor baseline (Radford et al., 2019), evaluated with the same updated pipeline. For the final analysis, we exclude tasks with $b_i - c_i < 0.5$: a reference below chance would reward lower accuracy, while a reference too close to chance makes normalization unstable. This excludes EWoK and Entity Tracking ($T=5$); both remain in the per-task analysis. Normalized scores are capped at ± 2 when $b_i - c_i < 5$ (binds for Reading-SPR). A score of 1.0 ties the updated anchor on the included tasks.⁴ Development and submission selection used the original six-task aggregate with pre-update reference scores; the final analysis reported here uses this revised five-task definition.

4 Results

The auxiliary changes no task significantly. To isolate the auxiliary’s effect we run a single-axis ablation: the identical recipe (Table 1) trained with the auxiliary weight set to $\lambda=0.1$ (aux) and to $\lambda=0$ (no-aux), six seeds each. Table 2 reports the seed-averaged per-task scores, the chance-anchored aggregate of Eq. 3, and the paired difference. No task reaches significance. The aggregate moves $+0.03$ ($p=0.29$) and Entity Tracking $+0.1$ ($p=0.89$), and the largest effect is a slight EWoK *drop* ($p=0.11$). Per-seed differences are the size of seed noise and change sign across seeds. The auxiliary therefore

⁴The reference scores are BLiMP 65.37, Supplement 59.60, COMPS 51.81, Reading-SPR 1.64, and Reading-ET 9.63.

Hyperparameter	Value
Architecture	LTG-BERT encoder
Training scheme	GPT-BERT hybrid
Shape (L/H/d/FFN)	12 / 6 / 384 / 1280
Parameters	33.4M (36.2M exported)
Vocabulary	16,384 (BPE)
Objective mix	93.75% MNTP / 6.25% CLM
Auxiliary	data2vec EMA self-distillation
Aux. λ / EMA τ	0.1 / 0.999
Aux. target	smooth- L_1 , top- $K=8$ layers
Optimizer / LR	LAMB / 7×10^{-3}
LR scheduler	cosine, warmup/cooldown 1.6%
Precision / batch	bf16 / 128 sequences
Steps / epochs	9,656 / 10
Words seen	99.95M (cap: 100M)
Masking	spans 1–3, rate 0.3 $\rightarrow$ 0.15
Seq. length	128 $\rightarrow$ 256 $\rightarrow$ 512
Seed (submission)	3
Compute	1 $\times$ L40S, 2.3 h, $\approx 3.2 \times 10^{16}$ FLOPs

Table 1: Training recipe for the submitted model (exp_seed_d2v_s3).

Task	Chance	Aux ($\lambda=0.1$)	No-aux ($\lambda=0$)	Δ	p
aggregate	0	1.369	1.342	+0.03	0.29
BLiMP	50	72.68	72.61	+0.07	0.75
Supplement	50	59.40	58.33	+1.07	0.21
EWoK	50	50.38	51.77	−1.39	0.11
Entity Tracking	20	21.38	21.26	+0.12	0.89
COMPS	50	52.23	52.20	+0.03	0.86
Reading (SPR)	0	4.93	4.81	+0.12	0.15
Reading (eye-tr.)	0	11.13	11.12	+0.01	0.97

Table 2: Seed-controlled ablation of the data2vec auxiliary: mean over six matched seeds, auxiliary toggled $\lambda=0.1$ (aux) vs. $\lambda=0$ (no-aux). Chance is the per-task chance level; the aggregate follows Eq. 3, excluding EWoK and Entity Tracking (0 is chance, 1 ties the updated GPT-2 anchor on the included tasks). Δ is the mean aux–no-aux difference, p a paired t -test. No task reaches significance.

provides no measurable benefit at the 10M-word scale. Both aux and no-aux sit above the re-scored GPT-2 anchor on Entity Tracking (19.10, itself below the 20.0 chance level), so the above-chance behavior is present with or without the auxiliary. Using the original development aggregate gives the same qualitative conclusion (+0.03, $p=0.67$).

4.1 The development gain was a single-seed artifact

The study was motivated by a striking development result: before the exposure cap, one configuration reached Entity Tracking ≈ 28 , far above its no-auxiliary counterpart. Two facts dissolve it. First, at this scale the seed matters more than the auxiliary. Table 3 reports the six matched seed

Seed	Aux ($\lambda=0.1$)	No-aux ($\lambda=0$)	Δ
0	20.76	23.06	−2.30
1	21.23	23.14	−1.91
2	21.54	21.28	+0.26
3	21.53	20.49	+1.04
4	20.04	19.67	+0.37
5	23.18	19.93	+3.25
mean	21.38	21.26	+0.12
sd	1.04	1.53	2.03

Table 3: Per-seed Entity Tracking for the six matched seed pairs behind Table 2, reported to compare the size of the auxiliary effect with the size of seed variation. The per-seed effect is negative for two seeds and positive for four, and its standard deviation (2.0) is an order of magnitude larger than its mean (+0.1); the paired difference is not significant ($p=0.89$).

pairs individually: the per-seed aux–no-aux difference swings from -2.3 to $+3.2$, is negative for two seeds and positive for four, and its standard deviation (2.0) is an order of magnitude larger than its mean (+0.1). The mean effect of the auxiliary is small relative to this seed-to-seed spread, so a comparison that rests on one seed per condition, as our development comparison did, can be dominated by noise. The development gap was moreover measured on the pre-update pipeline, where per-seed Entity Tracking ranged more widely (some seeds falling to 15–17); comparing a lucky auxiliary seed to an unlucky baseline seed there manufactures a large gap from noise. Second, the pre-update evaluation pipeline scored Entity Tracking differently (the July 2026 update filters $\approx 29\%$ of items; see Limitations), inflating the apparent margin further. Re-evaluating for both, with matched seeds and one pipeline, the gap vanishes (Table 2). The submitted model (data2vec seed 3) was selected using the pre-update aggregate; it is a strong entry, but its strength is not attributable to the auxiliary.

4.2 Possible explanations and attempts to improve the auxiliary

Two structural properties may explain why the auxiliary fails. (i) The auxiliary regresses the student’s projected state onto the EMA teacher’s contextual embedding *at the same masked positions* that masked prediction already supervises with a cross-entropy target. Predicting a slowly-averaged copy of one’s own representation of the masked token may largely restate the masked-prediction objective. (ii) A sequence-length guard disables the auxiliary once inputs reach the full 512-token length (a memory measure), so it is turned off for the final

Variant	Change	Entity Tr.
No-aux (baseline)	$\lambda=0$	19.7–23.1
Aux (default)	$\lambda=0.1$	20.0–23.2
Higher weight	$\lambda=0.15$	21.7–22.6
Decoupled mask	disjoint, rate 0.40	20.3–22.9
Wider target	± 4 -token window	22.2
Slower teacher	$\tau \rightarrow 0.9999$	19.1

Table 4: Single-axis attempts to rescue the auxiliary. None of the four modifications exceeds the best no-auxiliary Entity-Tracking score.

refinement phase of training (i.e., the phase that most shapes the exported weights).

We retrained the recipe with one change at a time, each targeting a suspected cause: a higher weight ($\lambda=0.15$), a decoupled higher-rate auxiliary mask, a ± 4 -token target window, and a slower teacher ($\tau \rightarrow 0.9999$). None helped. Entity-Tracking scores for the four modifications ranged from 19.1 to 22.9, compared with 19.7–23.1 for the no-auxiliary baseline (Table 4). Within the range we tested, the negative result is a property of the objective at this scale, not of a single unlucky hyperparameter.

5 Conclusion

A low-weight data2vec-style EMA self-distillation auxiliary does not improve a 33M-parameter hybrid-trained LTG-BERT at the BabyLM-2026 Strict-Small scale. In a seed-controlled, single-axis ablation the auxiliary moves no zero-shot task significantly (aggregate $+0.03$, $p=0.29$).

The large Entity-Tracking gain we first observed was a favourable seed draw scored on the pre-update pipeline. Four single-axis variants of the auxiliary (weight, mask, target window, teacher rate) did not change the outcome. The model itself remains a competitive Strict-Small entry, but we attribute that competitiveness to the hybrid training recipe and EMA weight-averaging, both already present in the no-auxiliary baseline, rather than to the auxiliary. We hypothesize that, at this scale, latent prediction would need to provide more complementary supervision to improve on the hybrid baseline.

Limitations

Unreproduced baseline score. We could not reproduce the Entity-Tracking score published on the GPT-BERT-mixed baseline’s model card (39.95 under MNTP scoring): our retraining of the published recipe reaches 20.1 under the official 2026 pipeline. All Entity-Tracking claims in this paper are there-

fore within-pipeline comparisons against our own no-auxiliary baseline, and we cannot rule out that the published number reflects a configuration detail we failed to replicate.

Seed and statistical power. Our conclusion rests on six matched seed pairs. We found no statistically significant improvement, but this comparison has limited power to detect small effects. The submitted model was selected from the auxiliary seeds using the pre-update aggregate, a selection that inflates its pre-update headline number but is orthogonal to the aux-vs-no-aux comparison, which is seed-matched.

Evaluation pipeline update. A July 2026 update to the official pipeline changed the Entity-Tracking data selection. Re-scored under the updated pipeline, the submitted model’s full-eval Entity-Tracking average moves from 21.66 to 19.92 (regular subset 23.28 $\rightarrow$ 21.53) and the GPT-2 anchor from 21.68 to 19.10 (regular subset). EWoK also changes slightly (53.41 $\rightarrow$ 51.82, full evaluation) from an unrelated fix in the same update. EWoK remains above chance under both pipeline versions. EWoK and Entity Tracking are excluded from the updated aggregate because their updated GPT-2 reference scores are below chance (Section 3). The updated GPT-2 anchor’s regular-only and full Entity-Tracking scores both round to 19.10.

On the updated pipeline the submitted model still scores above the re-scored anchor on Entity Tracking (21.53 vs 19.10 regular; 19.92 vs 19.10 over all three categories), so the within-pipeline comparison that our analysis rests on is preserved. The absolute margin, however, is smaller than under the pre-update pipeline and sits close to the 20.0 chance level on the three-category average, which we flag as a caveat on the strength of the Entity-Tracking result.

Acknowledgments

We are grateful to our instructor Zdeněk Kasner, whose course on neural language models at Czech Technical University in Prague introduced us to the BabyLM Challenge and whose guidance and editorial advice made this submission possible.

References

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann

- LeCun, and Nicolas Ballas. 2023. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli. 2023. Efficient self-supervised learning with contextualized target representations for vision, speech and language. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.
- Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. 2022. data2vec: A general framework for self-supervised learning in speech, vision and language. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*.
- Adrien Bardes, Jean Ponce, and Yann LeCun. 2022. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*.
- Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every Layer Counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 238–252.
- Lucas Georges Gabriel Charpentier and David Samuel. 2024. GPT or BERT: why not both? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 262–283.
- Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*.
- Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 English sentences, aligned with existing EEG and reading time data. *Behavior Research Methods*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.
- Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. 2020. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Manu Srinath Halvagal, Axel Laborieux, and Friedemann Zenke. 2023. Implicit variance regularization in non-contrastive SSL. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Hai Huang, Yann LeCun, and Randall Balestriero. 2025. LLM-JEPA: Large language models meet joint embedding predictive architectures. *arXiv preprint arXiv:2509.14252*.
- Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, and 1 others. 2025. Elements of world knowledge (EWoK): A cognition-inspired framework for evaluating basic world knowledge in language models. *Transactions of the Association for Computational Linguistics*, 13.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics (TACL)*, 8.
- Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.
- Shentong Mo and Shengbang Tong. 2024. Connecting joint-embedding predictive architecture with contrastive self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Technical Report*.
- David Samuel. 2023. Mean BERTs make erratic language teachers: the effectiveness of latent bootstrapping in low-resource settings. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 221–237.
- David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974.

Task	Fast	Full
BLiMP	72.60	72.07
Supplement	60.40	60.05
EWoK	53.23	53.41 (51.82 after change)
Entity Tracking	23.28	21.66 (19.92 after change)
COMPS	51.85	51.85
Reading (SPR)	4.77	4.77
Reading (eye-tr.)	11.30	11.30
Reading overall	8.03	8.03

Table 5: Fast versus full zero-shot evaluation of the submitted model (MNTTP scoring). Fast Entity Tracking is the regular subset; full is the average over all three categories. Unparenthesized values use the pre-update pipeline. Parenthetical values are the post-update-pipeline scores (July 2026): the Entity-Tracking data filter and an unrelated EWoK scoring fix each move one task.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics (TACL)*, 8.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

A Full-Evaluation Results

The leaderboard submission is produced by the full evaluation pipeline, which differs from the fast evaluation used in the body of the paper. Table 5 compares the two conventions for the submitted model. Under the pre-update pipeline, its Entity-Tracking scores are 23.28 (regular), 23.96 (move-contents), and 17.76 (ambiguous-reference); the ambiguous-reference category is thus the one that pulls the full-eval average below the regular subset. By linguistic domain, BLiMP decomposes into morphology 81.7, syntax 68.9, syntax–semantics 69.3, and semantics 66.4. Table 6 reports the full pipeline’s (Super)GLUE fine-tuning results.

Task	Accuracy	F1	MCC
BoolQ	0.673	0.744	0.294
MultiRC	0.659	0.453	0.288
RTE	0.547	0.496	0.085
WSC	0.712	0.595	0.376
MRPC	0.711	0.825	0.255
QQP	0.722	0.625	0.404
MNLI	0.469	—	—

Table 6: (Super)GLUE fine-tuning results of the submitted model under the full evaluation pipeline (accuracy, F1, Matthews correlation).

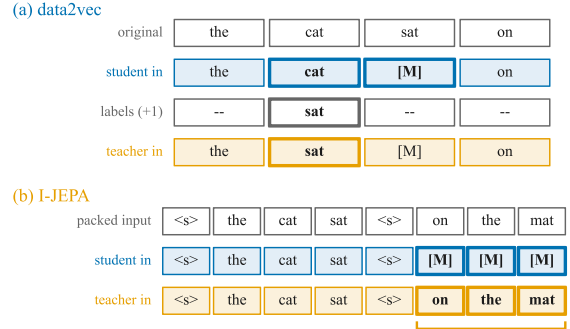


Figure 2: What each network sees (toy inputs). (a) data2vec variant: spans of 1–3 tokens are masked; with the MNTTP label shift, the position of “cat” predicts “sat”, and the teacher’s restored view carries the target token at that prediction position. (b) I-JEPA variant: one whole <s>-delimited line of the packed (restored) input is masked in a fresh copy; the bracket marks the masked line, which is predicted as a single pooled vector.

B Auxiliary Mechanics and the I-JEPA Variant

Both variants share the EMA teacher of Eq. 1 (Section 3).

What each network sees. Figure 2 shows the two auxiliary branches. In the **data2vec** variant (a), spans of 1–3 tokens are masked and, because labels are shifted by one position (MNTTP), the position of “cat” predicts “sat”; the auxiliary regresses the projected student state Wh_t at that position onto the teacher’s state $\bar{z}_t$, the mean of the teacher’s layer-normalized top 8 layers. The teacher’s restored view places the target token at the prediction position rather than rebuilding the original sequence exactly (Section 3). In the **I-JEPA** variant (b), one whole <s>-delimited corpus line (≥ 10 tokens) is masked in a fresh copy of the restored input; the student’s states over the masked line are mean-pooled, passed through a two-layer MLP head, and regressed with a cosine loss onto the teacher’s mean-pooled final-layer representation of the same line, for at most 8 sequences per batch.

The variants share the EMA teacher, the weight $\lambda=0.1$, the 0.5 loss-dropout, and the 512 cutoff; they differ in masking granularity (short spans vs. a whole line), target construction (layer-normalized top-8 mean per token vs. final-layer mean-pool per segment), prediction head (single linear vs. two-layer MLP), and loss (Huber vs. cosine).