

Causal, Hybrid, and Ternary Transformers under a Restricted Data Budget

Arnav Srivastav

University of California, Santa Cruz
Santa Cruz, CA 95064, USA
asriva31@ucsc.edu

Abstract

We evaluate five decoder-only Transformers trained in the BabyLM 2026 Strict-Small setting. In a closely matched training comparison, Causal-30M and Hybrid-30M share architecture, initialization, data, tokenization, optimization settings, seed, and training exposure. Causal-30M reaches lower held-out causal next-token loss. Their benchmark results come from different evaluation pipelines—server-held-out evaluation for Causal-30M and preliminary local evaluation for Hybrid-30M—so we report them descriptively and do not treat their benchmark differences as an objective-level ranking. A larger 64.57M-parameter hybrid configuration improves the preliminary NLP Average and substantially improves Entity Tracking, but its gains are not uniform across evaluation categories.

Separately, we compare architecture- and exposure-matched dense and fake-quantized ternary models. In this single-seed comparison, Ternary-30M averages 42.90 across the eight categories measured for both models, against 43.45 for Dense-30M, retaining 98.73% of the dense model’s measured aggregate. The two models use different peak learning rates, so this result describes the tested training recipes, not the effect of ternarization alone. Overall, the experiments show benchmark-specific trade-offs across objectives, model configurations, and weight representations. The ternary experiment measures capability retention under fake quantization and does not assess packed storage or specialized low-bit inference.

1 Introduction

Progress in language modeling has been driven by scale: larger models, larger datasets, and more compute. Transformers made large-scale sequence modeling more parallelizable (Vaswani et al., 2017). Scaling studies described predictable relationships among language-modeling loss, parameter count, training data, and compute (Kaplan

et al., 2020; Hoffmann et al., 2022). Many contemporary modeling choices were developed and tested with billions or trillions of training tokens. When data is scarce, however, those choices may not transfer cleanly: additional capacity may be underused, alternative objectives may redistribute the available learning signal, and low-precision weight constraints may impose a greater optimization cost.

The BabyLM Challenge is a shared setting for studying these questions under a restricted pretraining budget. Its evaluation suite spans grammatical, conceptual, downstream, and developmentally motivated behavior, picking up differences that language-modeling loss alone can miss (Warstadt et al., 2025; Hu et al., 2024; Choshen et al., 2026). Previous BabyLM systems and related work on data-efficient pretraining have improved performance through architectural changes, knowledge distillation, shorter training sequences, and modified objectives (Samuel et al., 2023; Charpentier and Samuel, 2023; Timiryasov and Tastet, 2023; Press et al., 2021; Charpentier and Samuel, 2024).

GPT-BERT is especially relevant: it combines causal and masked language modeling in a single Transformer and outperformed its causal-only and masked-only variants in the 2024 BabyLM setting (Charpentier and Samuel, 2024). Whether hybrid training helps across other small-model configurations and evaluation dimensions is not yet settled.

We study five GPT-style, decoder-only Transformers in the BabyLM 2026 Strict-Small setting. The experiments fall into two groups. The first examines pretraining objective and model configuration. Causal-30M and Hybrid-30M each have 29.22M parameters and are matched in architecture, initialization, corpus, tokenizer, context length, optimization settings, seed, and training exposure. Causal-30M uses next-token prediction; Hybrid-30M alternates between causal and bidirectional masked-language-modeling updates. This comparison evaluates the complete hybrid recipe: attention

visibility, input corruption, prediction targets, and loss-bearing positions all differ, not just the loss function. We also evaluate Hybrid-65M, a larger hybrid configuration. It is wider and shallower than Hybrid-30M and uses different attention-head geometry, so it tests a different configuration rather than isolating parameter count.

The second group examines fake-quantized ternary training. Ternary networks represent forward-pass weights using three scaled levels (Li et al., 2016; Ma et al., 2024), while related BitNet work applies low-bit linear layers to Transformer language models (Wang et al., 2023). These methods are motivated by the potential to reduce storage and arithmetic costs.

We compare Dense-30M and Ternary-30M, two 29.66M-parameter causal Transformers matched in architecture, corpus, tokenizer, context length, seed, and training exposure. Ternary-30M ternarizes its attention and feed-forward projections during the forward pass but keeps full-precision latent weights for optimization. This experiment measures capability retention under a ternary constraint. It does not evaluate bit-packed storage or specialized low-bit inference.

1. In the most closely matched training comparison, Causal-30M reaches lower held-out causal next-token loss than Hybrid-30M. The models’ benchmark outputs came from different evaluation pipelines, so we treat their benchmark rows as provisional observations, not a controlled ranking of objectives.
2. Hybrid-65M improves on Hybrid-30M in validation loss, preliminary NLP Average, Entity Tracking, and several other evaluations, but the gains are uneven. Hybrid-65M also differs in width, depth, and attention-head geometry. This comparison describes a larger configuration, not an isolated effect of parameter count.
3. In a single architecture- and exposure-matched comparison, Ternary-30M retains 98.73% of Dense-30M’s mean across the eight categories measured for both models. The runs use different peak learning rates, so the result describes the tested dense and ternary training recipes rather than the effect of ternarization alone.

Together, these experiments show that training objective, model configuration, and weight repre-

sentation lead to different trade-offs across evaluation dimensions under a restricted data budget.

2 Related Work

Data-efficient pretraining. BabyLM provides a shared setting for comparing language models trained on developmentally plausible, restricted corpora (Warstadt et al., 2025; Hu et al., 2024). Previous results show that performance under these constraints depends on architecture and training procedure as well as parameter count (Warstadt et al., 2025; Hu et al., 2024). LTG-BERT optimized the Transformer architecture and pretraining recipe for a 100M-word corpus (Samuel et al., 2023), while ELC-BERT introduced learned connections to earlier layer outputs and won the original Strict and Strict-Small tracks (Charpentier and Samuel, 2023). Baby Llama instead used knowledge distillation from an ensemble trained on the same small corpus (Timiryasov and Tastet, 2023). Our work complements these system-level contributions by using matched model pairs to examine particular training and weight-representation choices rather than optimizing a single leaderboard submission.

Causal and hybrid objectives. Causal language modeling predicts the next token from left context. BERT-style masked modeling reconstructs corrupted inputs using bidirectional context (Devlin et al., 2018). GPT-BERT showed that causal and masked language-modeling objectives can share a single Transformer stack and beat its causal-only and masked-only variants in the 2024 BabyLM setting (Charpentier and Samuel, 2024). Our causal-hybrid comparison revisits this question with models matched in architecture, parameter count, corpus, tokenizer, context length, optimization settings, seed, and training exposure. We evaluate the complete hybrid recipe, not one component of masked training in isolation. Our procedure is an independent implementation, not an exact replication of GPT-BERT, so differing outcomes suggest that hybrid performance depends on the specific recipe and model configuration.

Low-bit and ternary models. Ternary Weight Networks introduced scaled three-level weights as an alternative to full-precision and binary networks, with potential reductions in storage and arithmetic cost (Li et al., 2016). More recent work has applied low-bit training to Transformers. BitNet introduced 1-bit BitLinear projections for language

Source	Objective/scale	Dense/ternary
SimpleWiki	3,622,739	3,624,923
Gutenberg	3,424,991	3,424,824
BNC Spoken	1,299,294	1,299,971
OpenSubtitles	1,199,672	1,199,986
Switchboard	217,758	217,758
Total	9,764,454	9,767,462

Table 1: Whitespace-delimited word counts for the two corpus generations, before end-of-text markers are inserted. The objective/scale and dense-ternary groups are compared only within their own corpus generations.

models (Wang et al., 2023), while BitNet b1.58 extended this approach to ternary weights and reported performance comparable to full-precision Transformers with the same model size and training exposure (Ma et al., 2024). These studies motivate our dense-ternary comparison, but our scope is narrower: we measure capability retention under fake-quantized ternary forward passes while retaining full-precision latent weights. We do not claim the storage, latency, or energy benefits of a packed low-bit deployment.

3 Data and Tokenization

3.1 Training corpora

We train all models on restricted natural English corpora assembled from five sources distributed with BabyLM 2026: Simple English Wikipedia, Project Gutenberg, BNC Spoken, OpenSubtitles, and Switchboard (Choshen et al., 2026). We exclude CHILDES, synthetic or teacher-generated text, and all BabyLM benchmark data from pre-training. This corpus-construction choice is not a claim that child-directed speech is unimportant. The objective and scale experiments—Causal-30M, Hybrid-30M, and Hybrid-65M—use the final corpus generation. The Dense-30M and Ternary-30M comparison uses a slightly earlier generation drawn from the same source mixture. Table 1 gives the composition of both corpora.

We normalize text with Unicode NFKC, then normalize common typographic punctuation, remove control characters and source metadata, and collapse whitespace. We extract text using source-specific processing rules, merge short units into medium-length blocks, and deduplicate at the block level after normalization. Blocks are shuffled with seed 1337 and split roughly 98/1/1 into training, validation, and test sets. No normalized block appears in more than one split. Only the training split

goes to the optimizer; the validation split measures pretraining loss, and the test split is held out from both training and model selection.

The final corpus has 9,799,922 whitespace-delimited words after end-of-text markers are inserted. The optimizer split gets 9,607,048 words, with 99,498 and 93,376 words set aside for validation and testing. The earlier dense-ternary corpus has 9,789,053 words total, 9,592,958 of them in the optimizer split. The two generations differ in corpus construction details, so we do not interpret performance differences between the two experimental groups as controlled effects.

3.2 Tokenization

We train byte-level variants of byte-pair encoding tokenizers from scratch on the training split of each corpus (Sennrich et al., 2015). This avoids relying on a tokenizer pretrained on additional text. In our tokenizer reports, no unknown tokens occur in the training, validation, or test splits. Both tokenizers preserve capitalization and punctuation and encode each end-of-text marker as a single token. They do not automatically insert beginning- or end-of-sequence tokens.

The objective and scale experiments use a vocabulary of 8,192 tokens. Their training split contains 14,421,277 BPE tokens, corresponding to 1.501 BPE tokens per whitespace-delimited word. The tokenizer reserves special tokens for padding, unknown inputs, end of text, masking, and sequence separation. The mask token is used during the masked-language-modeling steps of the hybrid objective.

The dense-ternary comparison uses a 16,384-token byte-level BPE tokenizer trained on its corresponding corpus generation. Its training split contains 13,336,022 BPE tokens, corresponding to 1.390 BPE tokens per whitespace-delimited word. This tokenizer reserves special tokens for padding, unknown inputs, and end of text. It does not reserve masking or separator tokens because both models in this comparison use causal language modeling.

The two tokenizer settings do not constitute a controlled vocabulary-size experiment. In addition to vocabulary size, the experimental groups differ in corpus generation, context length, dropout, batching, and model configuration. We therefore restrict causal comparisons to models that share the same corpus and tokenizer.

4 Models and Training

We train five GPT-style, single-stack Transformers with pre-LayerNorm residual blocks. Each model uses learned token and absolute positional embeddings, scaled dot-product self-attention, exact GELU activations, and feed-forward layers four times wider than the residual stream. The token embeddings and language-modeling head share their weights. Linear projections and the output head have no bias. LayerNorm has a learned scale but no bias.

We initialize embedding and linear weights from a normal distribution with standard deviation 0.02. For the output projections of the attention and feed-forward sublayers, we reduce the initialization standard deviation to $0.02/\sqrt{2L}$, where L is the number of Transformer layers. The five model configurations are listed in Table 2.

4.1 Training objectives

Causal-30M, Dense-30M, and Ternary-30M are trained with standard autoregressive language modeling. For a token sequence $x_{1:T}$, the objective is

$$\mathcal{L}_{\text{causal}} = -\frac{1}{T-1} \sum_{t=1}^{T-1} \log p(x_{t+1} | x_{\leq t}).$$

A causal attention mask prevents tokens from attending to later positions, and the loss is averaged over all valid next-token predictions.

Hybrid-30M and Hybrid-65M split their training between causal language modeling and bidirectional masked language modeling. The assignment is deterministic. Update s uses the masked objective when

$$(7919s \bmod 10000) < 5000.$$

Over 36,648 optimizer updates, this gives 18,322 masked updates and 18,326 causal updates.

On a masked update, each token other than padding, end-of-text, and mask tokens is selected independently with probability 0.15. If no token is chosen in a sequence that has eligible tokens, one eligible position is forced so that the sequence still contributes to the loss. Of the selected tokens, 80% are replaced by the mask token, 10% by a token sampled from the non-special vocabulary, and 10% are left unchanged (Devlin et al., 2018). Attention is bidirectional on these updates, and only the selected positions contribute to the loss.

This experiment compares causal training against the complete hybrid recipe. The hybrid condition changes the attention pattern, input corruption, prediction targets, and number of loss-bearing positions; it does not just swap the loss function. Causal-30M and Hybrid-30M otherwise share the same architecture, initial parameters, corpus, tokenizer, context length, optimizer, learning rate, seed, and total exposure.

The runs share a seed but are not paired minibatch-for-minibatch. Generating masked inputs draws extra random numbers, so the sequence of sampled training windows eventually diverges. We call the comparison seed- and exposure-matched, not example-order-matched.

4.2 Fake-quantized ternary training

Dense-30M and Ternary-30M share the same architecture, corpus, tokenizer, context length, batching configuration, seed, and training exposure. Architecturally, Ternary-30M differs by applying fake-quantized ternary weights to its attention and feed-forward projections (Li et al., 2016). The runs also use different peak learning rates, as reported below.

For each output channel i , we first compute the mean absolute magnitude of its full-precision latent weights:

$$m_i = \frac{1}{N_i} \sum_j |w_{ij}|.$$

The ternarization threshold is then set to

$$\Delta_i = 0.7m_i.$$

Weights at or below Δ_i in magnitude are mapped to zero. The remaining weights are represented by their sign and a channel-specific scale,

$$\alpha_i = \frac{\sum_j |w_{ij}| \mathbb{I}[|w_{ij}| > \Delta_i]}{\sum_j \mathbb{I}[|w_{ij}| > \Delta_i]},$$

with the denominator clamped to at least one. The forward-pass weight is

$$\tilde{w}_{ij} = \alpha_i \text{sign}(w_{ij}) \mathbb{I}[|w_{ij}| > \Delta_i],$$

which takes one of three values: $\{-\alpha_i, 0, +\alpha_i\}$. During backpropagation, a straight-through estimator passes gradients to the underlying full-precision latent weights (Bengio et al., 2013).

We apply this transformation to the combined query, key, and value projection, the attention output projection, and both feed-forward projections in

Model	Params	Layers	Width	Heads	MLP width	Context	Dropout	Objective
Causal-30M	29.22M	16	368	8	1,472	512	0.1	Causal LM
Hybrid-30M	29.22M	16	368	8	1,472	512	0.1	Causal + MLM
Hybrid-65M	64.57M	12	640	10	2,560	512	0.1	Causal + MLM
Dense-30M	29.66M	16	352	4	1,408	256	0.0	Causal LM
Ternary-30M	29.66M	16	352	4	1,408	256	0.0	Causal LM + ternary proj.

Table 2: Model configurations. Causal-30M and Hybrid-30M share the same architecture and parameter count. Dense-30M and Ternary-30M form a separate architecture-matched pair. Hybrid-65M is both wider and shallower than Hybrid-30M; it is a larger configuration, not a controlled change in parameter count alone.

every Transformer layer. Token and positional embeddings, LayerNorm parameters, activations, softmax operations, and the tied language-modeling head stay dense. Ternary-30M has 29,658,464 trainable parameters: 23,789,568 in ternarized projection matrices and 5,868,896 in dense components.

The ternary experiment measures how much model quality survives the forward-pass weight constraint. It does not assess a complete low-bit deployment. Training still uses full-precision latent weights and optimizer state, checkpoints are not bit-packed, and inference uses no specialized ternary kernels. We make no claims about deployment-level memory, latency, throughput, or energy savings.

4.3 Optimization

All models are trained with AdamW (Loshchilov and Hutter, 2019), using $\beta_1 = 0.9$, $\beta_2 = 0.95$, and gradient clipping at a global norm of 1.0. Weight decay of 0.1 applies to linear weight matrices. Embeddings, LayerNorm parameters, biases, and the tied output head do not receive weight decay.

The learning rate increases linearly during the first 2% of optimizer updates and then follows a cosine decay schedule, ending at 10% of its peak value. Training uses bfloat16 automatic mixed precision with full-precision parameters and optimizer state. We do not use gradient accumulation. Every run uses seed 1337. Table 3 summarizes the model-specific optimization and training-exposure settings.

Every 250 updates, we estimate validation loss on 20 batches sampled from the held-out validation split. Each configuration reuses the same validation windows at every evaluation. We use causal next-token loss as the common validation metric for all models; masked-language-modeling loss is excluded. This tests the same prediction task across configurations, but it is not objective-neutral:

Causal-30M optimizes causal prediction at every update, while the hybrid models do so on roughly half of theirs.

4.4 Training exposure

Training examples are contiguous token windows with starting positions sampled with replacement from the optimizer split. Table 3 summarizes the resulting training exposure. Causal-30M, Hybrid-30M, and Hybrid-65M use exact exposure counts reconstructed from the sampled token windows. Dense-30M and Ternary-30M use estimates based on the corpus-level words-per-token ratio.

All runs targeted a ceiling of 100M whitespace-word exposures. The optimizer splits contain fewer than 10M words, so this budget amounts to more than ten exposure-equivalent passes. Batches are also sampled with replacement rather than drawn as ordered sweeps through the corpus. We report both word exposure and exposure-equivalent passes and do not call the runs ten literal epochs.

Except for AoA, all benchmark results use the final checkpoints at roughly 100M exposures; we do not select checkpoints by minimum validation loss. Causal-30M, Hybrid-30M, and Hybrid-65M were trained on an NVIDIA H100 NVL. The metadata for the earlier Dense-30M and Ternary-30M runs did not record the exact GPU. The two experimental generations also differ in context length, tokenizer, dropout, and batching, so we do not compare their training times or memory consumption directly.

5 Evaluation

We use the BabyLM 2026 evaluation suite (Choshen et al., 2026). BLiMP and its Supplement test linguistic knowledge through minimal-pair judgments (Warstadt et al., 2019). EWOK tests physical and social world knowledge (Ivanova et al., 2024). COMPS tests conceptual properties and property inheritance (Misra et al., 2022). Entity

Model	Peak LR	Batch	Tok./step	Steps	BPE positions	Word exposures	Equiv. passes
Causal-30M	3×10^{-4}	8	4,096	36,648	150,110,208	99,984,450	10.407
Hybrid-30M	3×10^{-4}	8	4,096	36,648	150,110,208	99,987,918	10.408
Hybrid-65M	3×10^{-4}	8	4,096	36,648	150,110,208	99,987,918	10.408
Dense-30M	3×10^{-4}	32	8,192	16,970	139,018,240	99,999,545 [†]	10.424 [†]
Ternary-30M	2×10^{-4}	32	8,192	16,970	139,018,240	99,999,545 [†]	10.424 [†]

Table 3: Optimization and training-exposure settings. Word-exposure counts for Causal-30M, Hybrid-30M, and Hybrid-65M are reconstructed exactly from the sampled token windows. Values marked with [†] are estimates based on the corpus-level words-per-token ratio. Dense-30M and Ternary-30M use different peak learning rates, so their comparison does not isolate weight representation alone.

Tracking evaluates whether models track changing information about entities across discourse, while GlobalPIQA tests cultural commonsense reasoning. The (Super)GLUE category summarizes seven fine-tuning tasks. Reading measures how much model surprisal improves the prediction of human reading times. Age of Acquisition (AoA) compares word-learning trajectories across training with human acquisition data.

Aggregation. Let $\mathcal{N}$ be the seven conventional NLP categories: BLiMP, BLiMP Supplement, EWoK, Entity Tracking, COMPS, GlobalPIQA, and (Super)GLUE. The reported aggregates are

$$\begin{aligned} \text{NLP} &= \frac{1}{7} \sum_{c \in \mathcal{N}} s_c, \\ \text{HumanLike} &= \frac{s_{\text{Reading}} + s_{\text{AoA}}}{2}, \\ \text{Overall} &= \frac{\sum_{c \in \mathcal{N}} s_c + s_{\text{Reading}} + s_{\text{AoA}}}{9}. \end{aligned}$$

The aggregation script assigns zero to missing categories. AoA is unavailable for Dense-30M and Ternary-30M, so their Overall scores are incomplete and cannot be compared directly with complete nine-category scores. For this pair, we use the mean over the eight available categories as the primary aggregate and report the zero-filled Overall score only to match the leaderboard calculation.

Checkpoints and provenance. Final-checkpoint evaluations use the last saved state of each run rather than a checkpoint selected according to validation loss or downstream performance. AoA uses 19 checkpoints distributed across pretraining and is available only for Causal-30M, Hybrid-30M, and Hybrid-65M.

The reported results do not all have the same evaluation provenance. Causal-30M’s results come from the server-held-out evaluation. The Hybrid-30M and Hybrid-65M scores are preliminary local

aggregates assembled from artifacts spanning two evaluator revisions. Causal-30M and Hybrid-30M are closely matched at the training level, but their benchmark results were not produced by a common evaluation pipeline. Comparisons between the server and local scores are therefore descriptive rather than definitive leaderboard rankings.

6 Results

Table 4 reports the available evaluation results. Causal-30M, Hybrid-30M, and Hybrid-65M make up the objective and model-configuration comparison at the training level, although their reported benchmark results do not share the same evaluation provenance. Dense-30M and Ternary-30M form a separate architecture- and evaluation-matched comparison using a different corpus generation, tokenizer, and context length. We interpret results within each experimental group. Cross-group differences are not controlled effects.

6.1 Causal versus hybrid training

Causal-30M and Hybrid-30M are our most closely matched training comparison. They share architecture, parameter count, initialization, corpus, tokenizer, context length, optimization settings, seed, and total word exposure. Their intended experimental difference is the objective recipe: Causal-30M uses causal language modeling on every update, whereas Hybrid-30M alternates between causal and masked-language-modeling updates. The hybrid condition also changes attention visibility, input corruption, prediction targets, and the number of loss-bearing positions.

On the common held-out causal next-token metric, Causal-30M reaches a final validation loss of 3.5421, compared with 3.7891 for Hybrid-30M. The tested causal recipe therefore produces better held-out causal next-token predictions in this run. This is not an objective-neutral measure of general

Model	BLiMP	Supp.	EWoK	Entity	COMPS	GPIQA	(S)GLUE	Read.	AoA	NLP	Human	Overall
<i>Objective and model-configuration comparison: 8k BPE, context 512</i>												
Causal-30M	66.04	52.79	50.95	19.97	51.93	35.64	64.20	0.63	17.96	48.79	9.29	40.01
Hybrid-30M [†]	63.73	50.03	51.00	16.46	51.23	36.64	64.53	0.73	-11.83	47.66	-5.55	35.83
Hybrid-65M [†]	65.27	51.87	50.51	31.30	51.64	34.67	65.82	0.37	3.54	50.15	1.95	39.44
<i>Dense-ternary comparison: 16k BPE, context 256</i>												
Dense-30M	68.33	55.61	52.23	19.93	51.70	36.62	62.97	0.19	–	49.63	–	38.62[‡]
Ternary-30M	65.90	54.63	52.17	20.45	51.63	35.58	62.62	0.18	–	49.00	–	38.13 [‡]

Table 4: Available BabyLM evaluation results. Causal-30M was evaluated using the server-held-out pipeline. Results marked with [†] are preliminary local results assembled from evaluation artifacts spanning two evaluator revisions. Dense-30M and Ternary-30M were not evaluated on AoA; their displayed Overall scores assign zero to the missing category and are incomplete aggregates ([‡]). Boldface identifies the numerically higher value within the evaluation-matched dense-ternary comparison and does not indicate statistical significance.

model quality, since causal prediction is optimized on every Causal-30M update but only roughly half of Hybrid-30M’s updates.

The benchmark outputs did not come from a common evaluation pipeline. Causal-30M was evaluated through the server-held-out pipeline, whereas Hybrid-30M was evaluated locally using preliminary artifacts spanning two evaluator revisions. We report both rows but treat the numerical gap between their Overall scores as descriptive, not as a controlled causal-hybrid effect.

In the displayed results, Causal-30M is numerically higher on BLiMP, the BLiMP Supplement, Entity Tracking, COMPS, and AoA, whereas Hybrid-30M is numerically higher on EWoK, GlobalPIQA, (Super)GLUE, and Reading. These task-level patterns come from mismatched evaluation pipelines and are therefore descriptive. The preliminary Hybrid-30M AoA value of -11.83 accounts for most of the displayed Overall difference. This value has not been verified under a common frozen evaluator revision, so we treat it and aggregates derived from it as provisional.

6.2 Larger hybrid configuration

Hybrid-65M has 64.57M parameters, about 2.21 times as many as Hybrid-30M. It also differs in width, depth, and attention-head geometry, so this is a comparison between model configurations rather than a controlled change in parameter count alone.

Within the preliminary local evaluation outputs, the larger model raises the Overall aggregate from 35.83 to 39.44 and reduces final validation loss from 3.7891 to 3.6122. The clearest gain is on Entity Tracking, which increases from 16.46 to 31.30. Hybrid-65M also scores higher on BLiMP, the BLiMP Supplement, COMPS, (Super)GLUE, and

AoA. Its NLP Average rises from 47.66 to 50.15, the highest displayed NLP Average among the three objective and model-configuration runs. Because both hybrid aggregates are assembled from preliminary local outputs, these differences should also be interpreted descriptively.

The gains are uneven. Hybrid-65M scores below Hybrid-30M on EWoK, GlobalPIQA, and Reading. Its preliminary Overall value is numerically below Causal-30M’s server-reported value, although their unmatched evaluation provenance prevents interpreting this difference as a ranking. Causal-30M has a Human-like Average of 9.29, compared with 1.95 for Hybrid-65M.

As a sensitivity calculation, excluding AoA gives means of approximately 43.93 for Hybrid-65M and 42.77 for Causal-30M over the remaining eight displayed categories. The ordering of the displayed aggregates reverses when AoA is removed. The two rows have different evaluation provenance, so this calculation shows how sensitive the aggregate is to AoA. It does not establish an evaluation-matched ranking between the models.

6.3 Dense versus ternary training

Dense-30M and Ternary-30M share the same architecture, parameter count, corpus, tokenizer, context length, batching configuration, seed, and training exposure. The main differences are the ternary forward-pass projections and the lower peak learning rate used for Ternary-30M.

AoA was not evaluated for either model. Across the eight categories measured for both, Dense-30M averages 43.45 and Ternary-30M averages 42.90. In this single comparison, the ternary configuration retains 98.73% of the dense model’s measured aggregate. Table 4 also reports the leaderboard-style Overall values obtained by assigning zero to

missing AoA, but we do not use those incomplete aggregates for comparison.

Dense-30M is numerically higher on seven of the eight available categories and has a lower final validation loss (3.9703 versus 4.0296). Ternary-30M is slightly higher on Entity Tracking. The models use different peak learning rates and only one seed, so the remaining gap cannot be attributed to ternarization alone.

7 Discussion

Causal and hybrid training. Causal-30M reaches lower held-out causal next-token loss than the closely matched Hybrid-30M configuration. In this run, the tested causal recipe produces better causal predictions, but the metric is not objective-neutral: Causal-30M optimizes causal prediction at every update, while Hybrid-30M does so on roughly half of them.

The displayed benchmark patterns are also mixed. Causal-30M is numerically higher on several categories, particularly AoA. Hybrid-30M is higher on EWoK, GlobalPIQA, (Super)GLUE, and Reading. The causal results are server-reported and the hybrid results are preliminary local outputs spanning two evaluator revisions, so these differences are descriptive, not evaluation-matched evidence of objective superiority.

Our experiment evaluates the complete hybrid recipe, not masked modeling in isolation. Hybrid training changes attention visibility, input corruption, prediction targets, and the number of loss-bearing positions all at once. Our procedure is also an independent implementation, not an exact replication of GPT-BERT. The differing outcomes suggest that hybrid performance depends on architecture, masking procedure, objective schedule, and evaluation protocol (Charpentier and Samuel, 2024; Hu et al., 2024).

Model configuration and aggregate sensitivity. Within the preliminary local evaluation, Hybrid-65M improves on Hybrid-30M in validation loss, NLP Average, Entity Tracking, and several other categories. It scores lower on EWoK, GlobalPIQA, and Reading. These differences cannot be attributed to parameter count alone: Hybrid-65M is also wider, shallower, and uses different attention-head geometry. The result therefore describes a larger hybrid configuration rather than an isolated scaling effect.

The displayed aggregate ranking is also sensitive to AoA. Hybrid-65M has the highest preliminary NLP Average, whereas Causal-30M has the highest reported Overall score. When AoA is excluded, their displayed ordering reverses. The two rows have unmatched evaluation provenance, so this does not establish a model ranking. Instead, it shows that Overall is strongly influenced by the developmental component. AoA is only one developmental measure, so these results do not support broader claims about cognitive plausibility.

Ternary capability retention. Ternary-30M remains numerically close to Dense-30M, retaining 98.73% of its measured eight-category aggregate. Dense-30M still has lower validation loss and higher scores on seven of the eight available categories. The models use different peak learning rates and only one seed, so the remaining gap cannot be attributed to ternarization alone.

This result is about capability retention under fake-quantized forward passes, not deployment efficiency. Training keeps full-precision latent weights, several components stay dense, and the implementation uses neither packed ternary storage nor specialized inference kernels. Establishing practical benefits will require matched optimization settings, multiple seeds, and a deployment-aware implementation.

8 Conclusion

We trained five compact Transformers under BabyLM Strict-Small. Causal-30M achieved the lowest causal validation loss, Hybrid-65M produced the highest preliminary NLP Average and sharply improved Entity Tracking, and Ternary-30M retained 98.73% of Dense-30M’s measured eight-category aggregate. Together, these results show that training objective, model configuration, and weight representation produce distinct strengths under restricted data.

9 Limitations

Our experiments have several limitations. First, each model configuration was trained with a single random seed, so we cannot estimate run-to-run variance or establish statistical significance. Although Causal-30M and Hybrid-30M are tightly matched in architecture, data, tokenizer, optimization, and training exposure, their reported benchmark results differ in provenance: Causal-30M was scored using the server-held-out evaluation, whereas the hybrid

scores are preliminary local aggregates. The exact magnitude of the reported performance difference should therefore be interpreted cautiously. The local hybrid results were assembled from artifacts spanning two evaluator revisions. In addition, the causal and hybrid runs are matched in seed and total exposure but not minibatch-for-minibatch: masked-input construction consumes additional random numbers, causing their sampled training windows to diverge. They are also matched in word exposure rather than in the number of loss-bearing prediction targets.

The Hybrid-30M and Hybrid-65M comparison does not isolate model scale. Hybrid-65M is simultaneously larger, wider, shallower, and configured with different attention-head geometry. Similarly, the dense-ternary comparison does not isolate weight representation completely because Ternary-30M uses a lower peak learning rate than Dense-30M. Age of Acquisition was not evaluated for either model in this pair, so their displayed Overall scores are incomplete aggregates that assign zero to the missing category. The two experimental groups also use different corpus generations, tokenizer vocabulary sizes, context lengths, dropout settings, and batching configurations; consequently, comparisons across groups are descriptive rather than controlled. The training corpora exclude CHILDES and therefore omit a major source of child-directed speech, limiting their developmental and domain diversity.

Our ternary experiment evaluates fake-quantized forward projections rather than an efficient deployment implementation. Training retains full-precision latent weights, several model components remain dense, checkpoints are not bit-packed, and no specialized ternary inference kernels were evaluated. We therefore make no empirical claims about storage reduction, inference latency, throughput, or energy consumption.

Finally, the exposure accounting is not equally precise across the two experimental generations. For Causal-30M, Hybrid-30M, and Hybrid-65M, we reconstructed the sampled token windows and obtained exact counts just below 100 million whitespace-word exposures. For Dense-30M and Ternary-30M, the reported value of 99,999,545 exposures comes from an earlier procedure that multiplies the processed-token count by the corpus-level words-per-token ratio. It should therefore be treated as an estimate rather than an exact word-by-word count. These exposure budgets correspond

to approximately 10.41–10.42 exposure-equivalent passes over the optimizer splits, not ten literal epochs, because examples are sampled with replacement.

Exact block-level deduplication was performed within each corpus generation, and no exact normalized-block overlap was found between its training, validation, and test splits. We did not, however, conduct an exhaustive semantic-overlap analysis against every evaluation item. These limitations restrict our conclusions to the configurations, seeds, and evaluation procedures examined here.

References

- Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. [Estimating or propagating gradients through stochastic neurons for conditional computation](#). *Preprint*, arXiv:1308.3432.
- Lucas Georges Gabriel Charpentier and David Samuel. 2023. [Not all layers are equally as important: Every layer counts BERT](#). *Preprint*, arXiv:2311.02265.
- Lucas Georges Gabriel Charpentier and David Samuel. 2024. [GPT or BERT: Why not both?](#) *Preprint*, arXiv:2410.24159.
- 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 and goes multilingual: Call for papers for the 2026 babylm workshop](#). *Preprint*, arXiv:2602.20092.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). *Preprint*, arXiv:1810.04805.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. [Training compute-optimal large language models](#). *Preprint*, arXiv:2203.15556.
- Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. [Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora](#). *Preprint*, arXiv:2412.05149.
- Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark,

- Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyürek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2024. [Elements of world knowledge \(EWOK\): A cognition-inspired framework for evaluating basic world knowledge in language models](#). *Preprint*, arXiv:2405.09605.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling laws for neural language models](#). *Preprint*, arXiv:2001.08361.
- Fengfu Li, Bin Liu, Xiaoxing Wang, Bo Zhang, and Junchi Yan. 2016. [Ternary weight networks](#). *Preprint*, arXiv:1605.04711.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). *Preprint*, arXiv:1711.05101.
- Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. 2024. [The era of 1-bit LLMs: All large language models are in 1.58 bits](#). *Preprint*, arXiv:2402.17764.
- Kanishka Misra, Julia Taylor Rayz, and Allyson Ettinger. 2022. [COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models](#). *Preprint*, arXiv:2210.01963.
- Ofir Press, Noah A. Smith, and Mike Lewis. 2021. [Shortformer: Better language modeling using shorter inputs](#). *Preprint*, arXiv:2012.15832.
- David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. [Trained on 100 million words and still in shape: BERT meets the British National Corpus](#). *Preprint*, arXiv:2303.09859.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. [Neural machine translation of rare words with subword units](#). *Preprint*, arXiv:1508.07909.
- Inar Timiryasov and Jean-Loup Tastet. 2023. [Baby llama: Knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty](#). *Preprint*, arXiv:2308.02019.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). *Preprint*, arXiv:1706.03762.
- Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. 2023. [BitNet: Scaling 1-bit transformers for large language models](#). *Preprint*, arXiv:2310.11453.
- Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjape, Adina Williams, Tal Linzen, and Ryan Cotterell. 2025. [Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora](#). *Preprint*, arXiv:2504.08165.
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2019. [BLiMP: The benchmark of linguistic minimal pairs for english](#). *Preprint*, arXiv:1912.00582.