

Some Assembly Required Pointer-Decoded Seq2Seq Models for German Without a Native Seq2Seq Backbone

Jan Pfister and Tom Völker and Andreas Hotho

Data Science Chair

Center for Artificial Intelligence and Data Science (CAIDAS)

Julius-Maximilians-Universität Würzburg (JMU)

{pfister,voelker,hotho}@informatik.uni-wuerzburg.de

Abstract

Pointer-decoded Seq2Seq transformers remain a strong recipe for aspect-based sentiment analysis (ABSA), but native German Seq2Seq backbones remain scarce, and current pretraining trends favor decoder-only architectures. Existing assembly recipes disagree on which pairings work. We therefore study models assembled from independently pretrained encoder and decoder LMs, primarily on German ABSA, where fixed pointer outputs expose both tuple accuracy and grammar violations. All nine backbone combinations train under both separate and parameter-sharing merged cross-attention, with separate attention slightly stronger overall. Depth and scale sweeps show that additional capacity pays off more on the encoder side for these pointer-decoded tasks, and our strongest two-stack assembly trails mBART but outperforms mT5-small on both German datasets. Motivated by decoder-only pretraining, we replace the encoder and decoder with one backbone that encodes the source and autoregressively produces pointer queries. Bidirectional source visibility and scale matter more consistently than pointer-query refinement: a 120M LLmle model exceeds both same-family two-stack means with roughly half the parameters, and a 1B ModernGBERT reaches mBART-level performance.

1 Introduction

Encoder-decoder Seq2Seq transformers remain the standard backbone for tasks that consume a full input before producing an output, such as machine translation (Tiedemann et al., 2024) and pointer-network structured prediction. In a pointer network, the decoder emits spans and labels by directly selecting *positions* in the encoded input (Yan et al., 2021). Because pointer decoding follows a fixed output grammar instead of generating a distribution over a subword vocabulary, comparably small models remain competitive with much

larger decoder-only LLMs. For example, in the evaluation of Pfister et al. (2025a), the pointer-decoded BARTABSA++ outperforms a GPT-4o fine-tune. Yet modern pretraining rarely produces encoder-decoder backbones from scratch (Zhang et al., 2025a), and native Seq2Seq models remain scarce for languages such as German (Pfister and Hotho, 2024). One alternative to pretraining from scratch is to assemble an encoder and decoder from separate pretrained transformer stacks: Rothe et al. (2020) build Vaswani et al.’s (2017) Seq2Seq from independently pretrained encoder/decoder LMs by inserting a cross-attention and fine-tuning.

Existing assembly recipes point in different directions. Rothe et al. (2020) achieve their best results when both sides initialize from an encoder, but leave assemblies that use decoders throughout or as encoders largely unprobed. BARTABSA++ later finds that scaling does not remove the performance ceiling in these configurations (Pfister et al., 2025a). T5Gemma (Zhang et al., 2025a) takes the opposite route, training a homogeneous two-decoder assembly through extensive further pretraining. On the same homogeneous configuration, T5Gemma2 (Zhang et al., 2025b) fuses self- and cross-attention into a single “merged attention” block to reduce parameters and ease initialization. However, it does not test merged attention when the encoder and decoder come from different backbones.

We study these assembly choices primarily on German aspect-based sentiment analysis (ABSA), where a fixed pointer output grammar makes model behavior easier to diagnose than open-ended generation metrics. Pointer-decoded Seq2Seq models are among the strongest approaches to ABSA (Yan et al., 2021; Wunderle et al., 2023; Pfister et al., 2025a). Prior German ABSA work therefore used multilingual BART as its backbone (Wunderle et al., 2023). However, strong native German encoder LMs (ModernGBERT) and decoder

LMs (LLaMmlein) are available in multiple sizes (Wunderle et al., 2026; Pfister et al., 2025b), opening assembly as a viable route to a native German Seq2Seq. A successful assembly is a usable artifact by itself: to our knowledge, the first pointer-decoded Seq2Seq for German ABSA built entirely from native components.

Current decoder-only pretraining makes it increasingly practical to use one backbone for both roles. We call this a *single-stack* model: the same transformer stack encodes the source and produces target pointer queries autoregressively without a dedicated encoder and decoder. Compared with a same-family encoder-decoder assembly, this design roughly halves the parameters, which is attractive for fine-grained analysis of large review collections (Ni et al., 2019). Merged attention also makes this reduction conceptually natural: it already projects source states and target history with the same key/value weights and normalizes their attention scores jointly. We therefore ask whether this remaining cross-attention boundary between the two model parts is necessary.

We organize this paper around four questions:

- RQ1:** Which encoder/decoder combinations train into a working pointer-decoded Seq2Seq for German ABSA, and which work best?
- RQ2:** How does T5Gemma2-style merged attention compare with separate cross-attention across homogeneous and heterogeneous assemblies?
- RQ3:** Where does the parameter budget belong, and can one backbone replace the two-stack assembly?
- RQ4:** How do assemblies compare against pre-trained Seq2Seq models like mBART, and does the recipe transfer across languages and tasks?

Our contributions are:

- (a) We compare T5Gemma2-style merged attention with separate cross-attention across all nine pairings in a grid of small German encoder and decoder backbones.
- (b) We build the first pointer-decoded Seq2Seq for German ABSA assembled entirely from native German components.
- (c) We show that encoder depth matters much more than decoder depth in our two-stack pointer models, and that a single LLaMmlein or ModernGBERT backbone can serve both roles.

- (d) We transfer the attention comparison to English ABSA and four further tasks, and the depth comparison to extraction and summarization.

We release the implementation, configurations, and result artifacts at <https://github.com/LSX-UniWue/some-assembly-required>.

 **Note:** Throughout the paper, we highlight interesting findings and insights we gained during the process in little boxes like this one.

2 Related Work

Pointer networks (Vinyals et al., 2015) emit pointers to indices from the input rather than tokens from a vocabulary, a natural fit for span-based tasks. BARTABSA (Yan et al., 2021) therefore applies this to aspect-based sentiment analysis (ABSA, example in section 3.1), on top of which BARTABSA++ (Pfister et al., 2025a) introduces three architectural enhancements. BARTABSA++ also report initial experiments combining this recipe with assembled Seq2Seq models (Rothe et al., 2020), but only with older English RoBERTa, BERT, and GPT2 backbones. Further, Wunderle et al. (2023) apply the pointer-network recipe with a multilingual mBART backbone on a German ABSA task (Wojatzki et al., 2017).

Rothe et al. (2020) warmstart both sides of a Seq2Seq from independently pretrained encoder and decoder LMs. A contemporaneous line of work on neural machine translation (NMT) leverages a pretrained encoder-only LM (BERT) on the source side, either by warmstarting the encoder (Imamura and Sumita, 2019) or by fusing BERT representations into the NMT model via attention (Zhu et al., 2020). T5Gemma (Zhang et al., 2025a) takes the opposite route, warm-starting both sides of its Seq2Seq from one pretrained decoder-only Gemma checkpoint and then continuing pretraining at length. T5Gemma2 (Zhang et al., 2025b) introduces *merged attention* on the same configuration to reduce parameters and ease initialization, motivated by the functional similarity of self- and cross-attention. Merged attention is in spirit related to multi-query (Shazeer, 2019) and grouped-query attention (Ainslie et al., 2023), although those share keys and values across heads of one attention block rather than between self- and cross-attention. Single-stack Seq2Seq masking has precedent in UniLM (Dong et al., 2019), which gives the source segment bidirectional visibility while keeping tar-

get prediction causal. Sun et al. (2025) recover an implicit pointing distribution from a decoder-only LLM’s existing attention and vocabulary distributions, without training a pointer layer, and combine it with ordinary vocabulary generation for query generation. Our model instead learns an explicit pointer head that outputs indices into the joint set of source positions and grammar symbols. We test this design with both decoder-pretrained LLäMmleins and encoder-pretrained ModernGBERT. We use mBART (Liu et al., 2020; Tang et al., 2020) and mT5 (Xue et al., 2021) as pretrained multilingual Seq2Seq baselines (section 5.4).

3 Methodology

3.1 Pointer-Decoded Seq2Seq for Aspect-Based Sentiment Analysis

We build on BARTABSA++ (Pfister et al., 2025a), a pointer network on top of a Seq2Seq backbone that emits sentiment triplets under a *fixed output grammar*. The encoder reads the input once, and the decoder then emits *pointers* autoregressively. At decoding step t , the model scores the decoder hidden state h_t^d against the n encoder token representations H^e and a set of special-token embeddings S , which contains EOS and task labels such as grammar delimiters and sentiment classes:

$$P_t = \text{softmax}([H^e; S] h_t^d). \quad (1)$$

P_t is the model’s entire output space: the decoder never generates over its subword vocabulary. A predicted pointer to an encoder position is dereferenced to that position’s encoder-token ID before it is fed back as the next decoder input; a special-token pointer feeds back the corresponding label or delimiter. BARTABSA++ extends the original BARTABSA (Yan et al., 2021) pointer head with three modifications to the inputs of eq. (1): feature normalization, parametrized gating, and an additional cross-attention block before the pointer head. We adopt all three throughout but do not analyze them further in this work, and defer to Appendix C.

Output grammar. Each sentiment triplet (like “Drivers” / “okay” / Positive in fig. 1) is serialized as a tuple of the form

`<ASP_BEGIN>  $a_s$   $a_e$  <OPN_BEGIN>  $o_s$   $o_e$  <SENT>`

where each span pointer is an index into the encoder positions, the holder block is optional, and <SENT> stands for one of <POS>, <NEU>, <NEG>. The full target sequence is a run of such tuples closed by EOS.

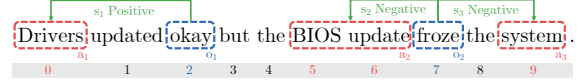


Figure 1: An example sentence for ABSA with aspect, opinion, sentiment, and word indices annotated. Reproduced from Pfister et al. (2025a), adapted from Yan et al. (2021).

We count a prediction as invalid if any completed tuple cannot be parsed under the task’s grammar. For ABSA, this includes a missing opinion marker or a span without a complete start/end pointer pair. For the SCARE sentence “*Ohne dieses neue update kann ich gar nichts mehr machen !*” (word indices 0–10) with gold aspect “update” (word 3), opinion “gar nichts mehr machen” (words 6–9), and negative polarity, the word-level rendering of the decoded pointer sequence is

`<ASP_BEGIN> 3 3 <OPN_BEGIN> 6 9 <NEG> EOS.`

Appendix B details how an emitted pointer is turned back into a decoder input embedding for the next decoding step.

3.2 Assembling a Seq2Seq from Pretrained Encoder and Decoder LMs

We assemble Seq2Seq models from three native German backbones: ModernGBERT, a native encoder; LLäMmleins, a native decoder; and LLäMmleins2Vec, an encoder converted from LLäMmleins using LLM2Vec (BehnamGhader et al., 2024). LLäMmleins2Vec shares its backbone weights with LLäMmleins but is adapted toward the encoder role. The pair therefore lets us compare a decoder used directly as an encoder with the same backbone adapted for encoding, alongside the dedicated ModernGBERT encoder. This requires extending Rothe et al.’s (2020) recipe, originally implemented only for (Ro)BERT(a) and GPT2, to more modern architectures such as LLaMA and ModernBERT.

To assemble a Seq2Seq from existing encoder and decoder models, we insert a cross-attention block into the decoder stack, and initialize the encoder and decoder weights from their respective checkpoints. To support arbitrary pairings, each checkpoint is adapted to its assigned side regardless of pretraining direction: encoder-side models run bidirectionally, while decoder-side models run causally and attend to the encoder states through the added cross-attention blocks. This yields both *homogeneous* configurations (initializing from the same backbone on both sides) and *heterogeneous*

ones (using a separate encoder and a dedicated decoder). Each has natural appeal: homogeneous for symmetry and (intuitively) “compatible” encoder/decoder representations from a single backbone, heterogeneous for letting each side use a backbone pretrained for its assigned role. When the two hidden sizes differ, our pointer scoring uses a learned linear projection from the encoder representation space (H^e and S) into the decoder hidden size. These projection parameters are trained jointly with the rest of the model.

Assembly also lets us choose the depth of each side independently of backbone choice and scale; neither side must retain its full pretrained stack. Because the pointer head scores encoder positions directly, intermediate encoder representations may already suffice or even be more suited for pointing. We therefore measure how performance changes when layers are removed from either side. In this *layer-cutoff sweep*, we independently retain the first N_e encoder layers and N_d decoder layers. We write $eN_e dN_d$ for each cell, e.g. e11d11 retains 11 layers on both sides.

3.3 Cross-Attention: Separate vs. Merged

The final two-stack design choice is how cross-attention connects the two pretrained sides. It is the only freshly inserted component inside each decoder layer, and we compare two ways to integrate it. A standard Seq2Seq decoder layer has two attention blocks that write into the residual stream in sequence: self-attention over the decoder states, then cross-attention to the encoder states, each with its own input norm and residual addition, following Vaswani et al. (2017). We call this the *separate* variant, the default in Rothe et al.’s (2020) assembly recipe, where the encoder-decoder attention is added as a second attention block to each pretrained decoder layer with freshly initialized projections.

T5Gemma2 (Zhang et al., 2025b) combines these blocks into one *merged attention* block with a single fused residual write (fig. 4). The decoder states H^d supply the queries, while the concatenated decoder and encoder states $X = [H^d; H^e]$ supply the keys and values. Suppressing head and positional-encoding notation, a single softmax over the joint self- and cross-attention logits produces one output, which is projected once and written once to the residual stream:

$$\text{softmax}\left(\frac{H^d W_Q (X W_K)^\top}{\sqrt{d_h}} + M\right) X W_V W_O, \quad (2)$$

where the mask M is causal within the self block and padding-only over the cross block. All four projections W_Q, W_K, W_V, W_O are reused unchanged from the decoder’s pretrained self-attention, so the merged block introduces no freshly initialized attention projection. T5Gemma2 presents this sharing as an efficiency tradeoff: fewer parameters and easier initialization at a slight quality reduction. Its experiments use homogeneous encoder-decoder pairs initialized from the same decoder-only checkpoint.

T5Gemma2 leaves open whether this parameter saving remains viable when source and target representations come from different backbones. We therefore compare *separate* and *merged* attention across both homogeneous and heterogeneous configurations in section 5.

3.4 A Single-Stack Pointer Model

Despite this shared attention operation, merged attention still computes source states in a separate encoder stack and target queries in the decoder stack. The single-stack model removes this remaining boundary by using one backbone for both roles, following the broader decoder-only pattern of conditioning and generation in the same stack. At step t , the backbone processes the source, a target start token, and the dereferenced pointer history of Appendix B as one sequence. Its first n hidden states replace the encoder representations H^e in eq. (1), and the current target state replaces the decoder query h_t^d . One backbone thus produces both source and query states, without an explicit encoder stack, encoder-to-decoder projection, or in-layer cross-attention.

We vary two factors. The first is source visibility: the *causal* mask lets source state i see only the source prefix through i , while the *prefix-visible* mask lets all source tokens see one another and keeps each target token causal over the full source and preceding target tokens, following UniLM (Dong et al., 2019). E.g. for decoder-pretrained LLäMmleIn, the causal condition therefore leaves its pretrained mask unchanged. The second factor is pointer-query refinement: the *enhanced* condition refines the target state with the BARTABSA++ pointer head of section 3.1, while the *direct* condition scores the backbone state at each time step against the candidates without this block.

4 Experimental Setup

4.1 Used Datasets

All five sentiment datasets annotate triplet structures with a target or aspect, an opinion, and a polarity class (fig. 1). For German we use SCARE (Sänger et al., 2016) (app-store reviews) and USAGE (Klinger and Cimiano, 2014) (product reviews), both with three-way polarity. The English transfer experiment (section 5.5) adds opener_en (Barnes et al., 2022), an English structured sentiment corpus with two-way polarity, and the two refined SemEval ABSA triplet sets (Xu et al., 2020) 14lap and 14res, the same English ABSA benchmark used by BARTABSA(++) (Yan et al., 2021; Pfister et al., 2025a). All five share the aspect/opinion/polarity triplet structure, so the pointer grammar and the tuple-F1 metric carry over unchanged across languages; opener_en additionally annotates opinion holders, which we emit as the optional third span block of the grammar (section 3.1) and which the tuple metric scores alongside aspect and opinion. Dataset statistics and one annotated example from each corpus are listed in Appendix E.

4.2 Backbone Models

We draw encoders and decoders from three native German backbones, each available in two sizes: LLäMmlein (120M, 1B), a native decoder (Pfister et al., 2025b); ModernGBERT (134M, 1B), a native encoder; and LLäMmlein2Vec (120M, 1B), the LLäMmlein decoder repurposed as an encoder (the latter two from Wunderle et al. (2026)). Any of the three can sit on either side of the assembly.¹

As pretrained Seq2Seq baselines we compare against the multilingual mBART-large-50 (610M) (Liu et al., 2020; Tang et al., 2020) and mT5-small (300M) (Xue et al., 2021).

4.3 Training and Evaluation

Since our aim is to conceptually establish whether/when these assemblies train and where the parameter budget belongs, not to extract maximal benchmark performance, we do not further

tune hyperparameters per configuration. We inherit the BARTABSA++ working hyperparameters throughout and build on its implementation (section 3.1), which exposes the three enhancements as toggleable components. The full hyperparameter list is in Appendix D.

The metric is weighted-overlap tuple F1 (Yan et al., 2021; Barnes et al., 2021). A predicted tuple counts as a true positive when its polarity matches the gold and each annotated span overlaps the corresponding gold span by at least one token. We weight that true positive by the average token-overlap fraction across its spans. This structured output makes evaluation unusually strict compared with open-ended generation metrics: predictions must follow the fixed grammar of section 3.1 and point to the relevant input spans to receive tuple-F1 credit.

The task-transfer experiments use DEFT (Spala et al., 2020), SRE (Gábor et al., 2018), German NER (GNER) on GermEval 2014 (Benikova et al., 2014), and CNN/DailyMail summarization (Nallapati et al., 2016), with three seeds per configuration. DEFT and SRE report pointer F1, GNER reports entity F1, and summarization reports ROUGE-1 (Lin, 2004).

For these three English tasks, we compare native BART-base (Lewis et al., 2020), homogeneous Llama-3.2-1B (Meta AI, 2024), and a heterogeneous pair of ModernBERT-large (Warner et al., 2025) and Llama-3.2-1B. For German GNER, we compare native mBART-large-50, homogeneous LLäMmlein-120M, and heterogeneous ModernGBERT-134M+LLäMmlein-120M. Due to data set size, GNER runs for one epoch, and for summarization, we use a fixed train/dev/test subset.

5 Results and Insights

5.1 How Much Does the Backbone Pairing Matter?

All three homogeneous and six heterogeneous backbone pairings produce working pointer models under both attention paths. That all nine work is notable because several cells use a backbone against its pretraining mask direction. In a heterogeneous cell, decoder queries must also attend to states produced by a different backbone: separate attention learns a dedicated block for this connection, whereas merged attention even reuses the decoder’s pretrained projections. To compare their quality, we screen each German backbone (ModernGBERT,

¹We name a configuration encoder2decoder, abbreviating ModernGBERT, LLäMmlein, and LLäMmlein2Vec as **mg**, **ll**, and **lv**: e.g. **mg2lv** is a ModernGBERT encoder with a LLäMmlein2Vec decoder, and **ll2ll** is LLäMmlein on both sides. The small backbone is the default; a 1B side is marked inline, as in **mg_{1B}2ll** (a 1B ModernGBERT encoder with a small LLäMmlein decoder).

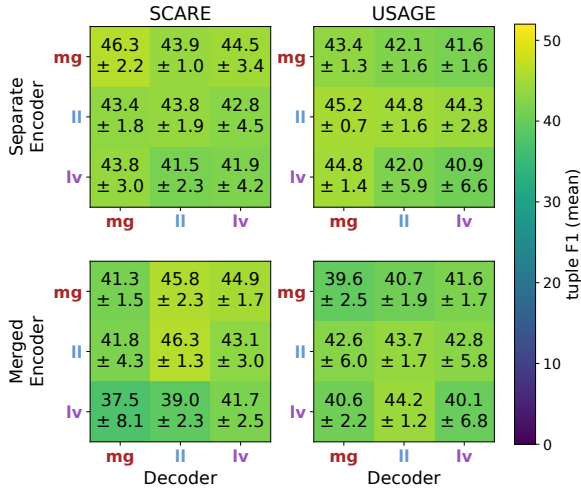


Figure 2: 3×3 encoder (rows) × decoder (columns) tuple F1 grid under separate (top) and merged (bottom) cross-attention, on SCARE (left) and USAGE (right). 5-seed mean ± std per cell.

LLäMmlein, LLäMmlein2Vec) as the encoder and decoder in a 3×3 grid at the small parameter size, under separate and merged cross-attention (RQ1, RQ2), with 5 seeds per cell.

Pairing affects quality, but no encoder row dominates across both datasets. Separate attention has the higher mean in 12 of 18 dataset/backbone cells and is 1.31 F1 stronger across all 90 matched dataset, backbone, and seed comparisons, with a paired 95% CI of [0.25, 2.37]. The clearest systematic loss appears when ModernGBERT serves as the decoder: merged attention lowers F1 by 3.93 on average, while its effect is near zero in the LLäMmlein and LLäMmlein2Vec decoder columns.

The invalid-output rates reveal the same decoder-side split. Across the six cells with LLäMmlein or LLäMmlein2Vec decoders, merged attention lowers the rate by 2.30 percentage points on average and does so in 55 of 60 matched runs. Across the three ModernGBERT-decoder cells, it instead raises the rate by 1.57 points, with increases in 21 of 30 runs.

The matched same-family LLäMmlein comparison used below shows the same benefit: merged attention does not improve mean F1 but lowers the mean invalid-output rate from 3.23% to 1.05% on SCARE and from 2.60% to 1.22% on USAGE.

Pairing affects quality, not viability: Every grid run trains under both separate and merged cross-attention, with separate attention slightly stronger overall.

encoder	decoder	SCARE F1	USAGE F1
120M	120M	46.26 ± 1.31	43.69 ± 1.69
1B	120M	49.30 ± 5.42	50.73 ± 1.17
120M	1B	41.41 ± 2.50	41.79 ± 6.50
1B	1B	43.61 ± 7.33	42.33 ± 5.68

Table 1: Full-depth LLäMmlein scale allocations under merged attention. Values are 5-seed test tuple-F1 mean ± std.

Appendix I documents why an earlier version of this paper reached a different conclusion and reports the corrected ablation.

5.2 Scaling: Where Should Parameters Go?

Two priors point to the encoder. Prior work reports that autoregressive NMT benefits from deep encoders and shallow decoders (Kasai et al., 2021), and the same encoder-heavy preference holds for pointer-decoded ABSA (Pfister et al., 2025a). The pointer head makes this preference plausible: it produces a softmax over encoder positions rather than tokens (eq. (1)), so the decoder “only” needs to deliver one query vector against H^e . We test this on two axes. **Depth** truncates each side to a prefix of its pretrained layers at fixed scale. **Scale** swaps a 120M and 1B checkpoint between the encoder and decoder while retaining full depth. The two-dimensional depth sweep appears in fig. 3, and table 1 compares the four full-depth scale allocations under merged attention.

The **depth** comparison is sharply asymmetric: encoder depth is essential, whereas much of the decoder depth is removable. Restoring the full encoder while holding the decoder at four layers adds 9.80 F1. Increasing the decoder from four to twelve layers while holding the encoder at four adds only 0.16 F1. The same asymmetry holds in both assemblies: restoring encoder depth adds 9.17 F1 for ll2ll and 10.44 for mg2ll, while adding decoder depth changes them by only −0.21 and +0.52 F1. *Decoder depth is largely free to cut, encoder depth is not.*

Even a full two-dimensional sweep on ll1B2ll under merged attention shows the same pattern (fig. 3). This sweep already pairs the larger 1B encoder with the smaller 120M decoder, yet after averaging each seed over the other depth axis, e22 exceeds e2 by 18.56 F1 on SCARE and 17.49 on USAGE. By comparison, d12 minus d2 is only 0.74 F1 on SCARE and 0.23 on USAGE.

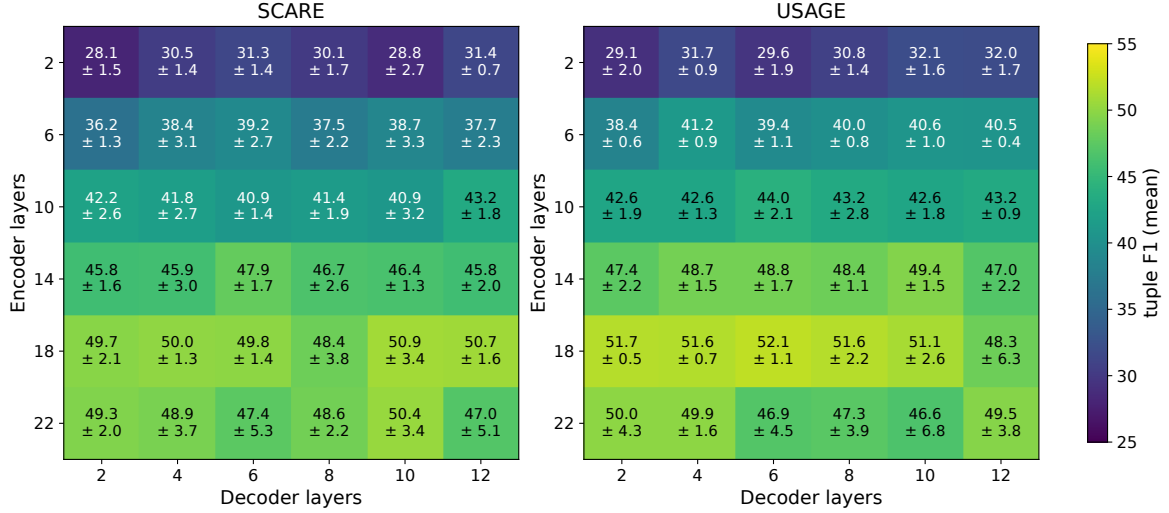


Figure 3: Aligned encoder $\times$ decoder depth sweep for a LLäMmleln-1B encoder and LLäMmleln-120M decoder under merged attention. Cells report 5-seed test tuple-F1 mean $\pm$ std. Performance is much more sensitive to encoder depth than decoder depth. The plots for the other pairings exhibit an identical pattern.

With separate attention, the full-depth **scale** swap points in the same direction but is much noisier. Pooling the ten comparisons gives the encoder-heavy allocation only a 4.31 F1 advantage on average, but only five seed-pairings are positive.

Under merged attention, the 1B-encoder/small-decoder allocation is even strongest in table 1. It beats the reverse allocation by 8.41 F1 across the seeds and beats the all-1B model by 7.05 F1 (Table 5).

 **Scale the encoder:** Restoring encoder depth yields a clear gain. Under merged attention, a 1B encoder with a small decoder also beats both the reverse allocation and the all-1B model.

We interpret this asymmetry through the pointer wiring of eq. (1): the decoder produces one query vector per step against H^e , so additional decoder capacity may saturate early, while richer encoder representations can keep improving discrimination among pointer candidates.

5.3 Can One Backbone Serve Both Roles?

Merged attention already weakens the boundary between encoder and decoder by placing source states and target history in one attention operation, but it still computes them in separate stacks. Since the scaling results also place most useful capacity on the encoder side, we ask whether one backbone can serve both roles.

For LLäMmleln-120M, both prefix-visible variants, in which all source tokens see one another, exceed both two-stack means with roughly half

their trainable parameters. Our paired comparisons support the prefix-visible enhanced model over the merged reference on both datasets and over the separate reference on SCARE. Across direct and enhanced pointer scoring, prefix visibility improves both datasets by 1.96 to 3.16 F1, while the enhanced refinement has no consistent advantage over direct scoring. The prefix-visible enhanced model also reduces median greedy-decoding latency by 40% to 44% relative to the merged attention two-stack model, and roughly halves peak test memory (table 13). The fully causal enhanced model also exceeds the merged means on both datasets and even the separate mean on SCARE. Scaling this causal model from 120M to 1B adds 2.58 F1 on SCARE and 7.06 on USAGE. Against a LLäMmleln-1B encoder with a 120M decoder, the causal single stack lies between the two attention paths: it leads separate by 2.61/5.74 F1 and trails merged by 2.85/0.32 on SCARE/USAGE. It exceeds the merged decoder-heavy and even all-1B means on both datasets. Against the all-1B model, the gains are 2.85/8.08 F1, but with roughly half the backbone parameters. At 1B, the full matrix shows the same pattern: prefix visibility raises each mean by 3.54 to 5.18 F1 across both query variants and datasets (tables 10 and 14). Both prefix-visible variants exceed the matched encoder-heavy two-stack means on both datasets, while direct and enhanced scoring differ by less than one F1. Across both LLäMmleln scales, source visibility is therefore

model	topology / access	params	SCARE F1	USAGE F1
<i>pretrained Seq2Seq baselines</i>				
mT5-small (enhanced)	native	300M	20.90 ± 1.79	23.91 ± 1.75
mBART-large-50	native	610M	56.70 ± 1.40	57.27 ± 1.69
<i>two-stack references</i>				
ll2ll	separate	$\sim 232\text{M}$	42.40 ± 3.07	44.57 ± 4.19
ll2ll	merged	203.8M	41.23 ± 2.55	42.83 ± 3.89
mg_{1B}2ll	merged	$\sim 1.1\text{B}$	50.78 ± 5.95	50.23 ± 8.21
<i>single-stack models</i>				
LLaMmlele-120M	causal	103.7M	43.89 ± 0.91	43.01 ± 1.06
LLaMmlele-120M	prefix-visible	103.7M	45.85 ± 1.80	46.05 ± 1.79
LLaMmlele-1B	causal	1.06B	46.45 ± 3.34	50.41 ± 1.59
LLaMmlele-1B	prefix-visible	1.06B	50.95 ± 1.82	53.95 ± 0.50
ModernGBERT-134M	causal	134M	32.35 ± 1.67	28.12 ± 2.24
ModernGBERT-134M	prefix-visible	134M	38.67 ± 2.10	37.98 ± 2.89
ModernGBERT-1B	causal	$\sim 1\text{B}$	39.84 ± 2.49	37.96 ± 3.27
ModernGBERT-1B	prefix-visible	$\sim 1\text{B}$	57.05 ± 1.64	56.82 ± 1.10

Table 2: Full-depth results: The LLaMmlele-120M single-stack and matched two-stack rows and the ModernGBERT-134M rows use 10 seeds; all other rows use 5. Values are test tuple-F1 mean $\pm$ std.

the clearest quality factor, while pointer-query refinement adds no consistent gain.

For ModernGBERT, prefix visibility adds 7.85 F1 at 134M and 17.38 at 1B when averaged over datasets and the direct and enhanced scoring variants. Scaling the ModernGBERT single stack from 134M to 1B improves every source-visibility and pointer-query configuration by 8.66 to 18.93 F1. The full factorial results and paired factor-effect intervals are in tables 11 and 12.

The 134M prefix-visible models remain below both small two-stack references. Scaling to 1B reverses that comparison: both prefix-visible pointer variants come within 1.83 F1 of mBART on each dataset. The enhanced model is also at least six F1 above both encoder-heavy two-stack means on each dataset. This connects the single-stack result to the earlier encoder-heavy finding: the strongest single-stack model uses the encoder-pretrained backbone to strongly encode the source and then also decode the pointers.

 **One backbone can be enough:** A strong encoder-pretrained backbone can encode the source and decode pointers in one stack: at 1B, prefix-visible enhanced ModernGBERT lies above both encoder-heavy two-stack means and reaches mBART-level F1.

5.4 Comparison to Pretrained Seq2Seq Baselines (RQ4)

Table 2 compares the full-depth models against mBART-large-50 (610M) and mT5-small (300M) under the same downstream training protocol. They

provide a calibration point for models whose encoder and decoder were pretrained jointly. Both encoder-heavy two-stack assemblies reach about 50 F1: [mg_{1B}2ll](#) trails mBART by 5.92/7.05 F1 on SCARE/USAGE, while [ll_{1B}2ll](#) trails it by 7.40/6.54 F1 (tables 1 and 2). The small [ll2ll](#) two-stack models also exceed mT5-small on both datasets despite using fewer parameters. At 103.7M parameters, the prefix-visible LLaMmlele single stack is about one third the size of mT5-small and exceeds it by 24.95/22.14 F1. At 1B, the prefix-visible LLaMmlele variants close much of the remaining gap but stay below mBART on both datasets. The 1B prefix-visible ModernGBERT single stack differs from mBART by only +0.35 F1 on SCARE and -0.45 on USAGE.

5.5 Transfer Across Languages

German ABSA is our primary setting, so we next change the language while retaining the rest of the setting (RQ2, RQ4). We repeat the attention comparison on 14lap, 14res, and opener_en (section 4.1), using homogeneous Llama-3.2-1B and heterogeneous ModernBERT-large+Llama-3.2-1B assemblies with 5 seeds per cell. The heterogeneous pair is the closest English analogue to [mg2ll](#), although its parameter allocation is decoder-heavy relative to section 5.2. Separate attention has the higher mean in five of six assembly/dataset comparisons. The heterogeneous gaps are small at 0.43, 1.33, and 0.65 F1, while the homogeneous effect varies more: separate leads by 1.67 F1 on 14lap and 4.69 on opener_en, but merged leads by 3.02

task	pair	native	separate	merged	Δ
DEFT	hom.	49.36 $\pm$ 0.70	35.94 $\pm$ 1.33	46.25 $\pm$ 1.15	+10.31
DEFT	het.	49.36 $\pm$ 0.70	38.77 $\pm$ 1.75	37.51 $\pm$ 1.28	-1.26
SRE	hom.	20.43 $\pm$ 1.14	3.22 $\pm$ 1.09	9.03 $\pm$ 3.74	+5.81
SRE	het.	20.43 $\pm$ 1.14	2.53 $\pm$ 1.87	3.07 $\pm$ 2.36	+0.54
GNER	hom.	83.31 $\pm$ 0.24	68.65 $\pm$ 0.50	69.98 $\pm$ 1.62	+1.33
GNER	het.	83.31 $\pm$ 0.24	45.12 $\pm$ 0.94	51.68 $\pm$ 0.90	+6.56
CNN/DM	hom.	32.35 $\pm$ 0.28	16.64 $\pm$ 1.13	24.04 $\pm$ 3.32	+7.40
CNN/DM	het.	32.35 $\pm$ 0.28	17.61 $\pm$ 1.90	15.06 $\pm$ 1.70	-2.54

Table 3: Task transfer with 3 seeds per cell. DEFT, SRE, and GNER report F1; CNN/DM reports ROUGE-1 $\times 100$. Δ is merged minus separate. Hom. and het. denote the homogeneous and heterogeneous model pairs specified in section 4.3.

on 14res. Thus, both homogeneous and heterogeneous assemblies train under either attention path in English as well, but neither closes the gap to the native BART and mBART references (full numbers in Appendix G).

5.6 Transfer Across Tasks

The language-transfer experiment changes the language while retaining sentiment extraction. We next ask whether the attention comparison transfers to English definition extraction (DEFT), English relation extraction (SRE), German NER (GNER), and English CNN/DailyMail summarization under the setup of section 4.3.

Merged attention raises six of eight means and all four homogeneous assemblies; among heterogeneous pairs it helps SRE and GNER but lowers DEFT and summarization. The largest gains occur for homogeneous DEFT (+10.31), homogeneous summarization (+7.40), and heterogeneous GNER (+6.56). On SRE and summarization, every assembly remains far below the native reference, even where merged attention helps. Across the 18 matched runs for the three structured transfer tasks, merged attention also lowers the invalid-output rate by 3.73 percentage points on average. The largest reductions are 20.40 points for homogeneous SRE, 4.04 for homogeneous DEFT, and 1.94 for heterogeneous GNER, each observed in all three seeds.

Beyond attention type, we test whether the encoder-heavy depth preference of section 5.2 transfers to these tasks. Encoder-heavy beats decoder-heavy in all 18 matched task, assembly, and seed comparisons, by 10.91 F1 on average with a paired 95% CI of [7.60, 14.21]. It has the highest mean in five of six groups; homogeneous SRE is the exception, where balanced reaches 8.76 F1 against 8.41 for encoder-heavy. Table 7 gives the complete means and standard deviations.

Interestingly, only summarization reverses the structured-task depth ranking. Separate attention stays near 16.7 ROUGE-1 across the three allocations, while merged rises from 16.90 $\pm$ 0.62 at e16/d4 to 29.64 $\pm$ 0.48 at e4/d16, approaching the native BART result of 32.35 $\pm$ 0.28. This suggests that the pointer-decoded models benefit more from encoder representation quality than free text generation encoder-decoder models. Likely because here the output vocabulary is much larger for the decoder to discriminate over, the decoder-heavy allocation pays off for summarization. The complete allocation results are in table 8.

6 Conclusion

Competitive pointer-decoded Seq2Seq models for languages where native Seq2Seq backbones are scarce can be assembled from independently pre-trained encoder and decoder LMs. In the 3×3 grid (RQ1, RQ2), both separate and merged attention train reliably, with separate attention slightly stronger overall. Encoder depth matters much more than decoder depth. At full depth, the 1B encoder with a small decoder also beats both the reverse allocation and the all-1B model under merged attention (RQ3). A single stack can replace the two-stack assembly: a prefix-visible LLäMmleln model exceeds both same-family two-stack means with roughly half the parameters, and a 1B prefix-visible enhanced ModernGBERT lies above both encoder-heavy two-stack means and reaches mBART-level performance. The strongest two-stack assembly trails mBART but clears mT5-small on both German datasets (RQ4). English ABSA also trains reliably under both attention variants, with separate attention usually stronger. Across broader task transfer, merged helps every homogeneous assembly; among heterogeneous ones, it helps SRE and GNER but lowers DEFT and summarization. Summarization also favors decoder-heavy depth.

Limitations

Scope of the empirical claim Our findings are anchored to pointer-decoded sentiment analysis on SCARE and USAGE in German and three English datasets, three German backbone families (ModernGBERT, LLäMmleln, LLäMmleln2Vec), and targeted transfer experiments on DEFT, SRE, GNER, and CNN/DailyMail. The encoder-heavy scaling rule (section 5.2) rests on the pointer-decoding wiring of eq. (1), where the decoder emits

a single query per step against encoder positions rather than generating over a subword vocabulary. The broader task-transfer cells use three seeds, and the English transfer and 1B single-stack cells use five. Absolute SRE and summarization scores are weak, and the summarization experiment uses a reduced 10,000/1,000/1,000 subset with a four-epoch recipe selected on development data. The full-depth width result is clearest for the same-family LLäMmlein comparison under merged attention.

No further pretraining of the assembled Seq2Seq. Unlike T5Gemma and T5Gemma2 (Zhang et al., 2025a,b), which continue pretraining the assembled Seq2Seq on a denoising objective before downstream finetuning, our assemblies go from initialization straight to the ABSA pointer task. The mBART comparison in section 5.4 is therefore not strictly like-for-like: mBART is a pretrained Seq2Seq with denoising-trained in-layer cross-attention, while our assemblies insert fresh cross-attention and train it only through the downstream signal. Further Seq2Seq pretraining might either widen or close the gap.

Acknowledgments

We thank Anton Vlasjuk for helpful discussions and for his master’s thesis work on related questions, which informed the early exploration of this project. We also thank the reviewers, especially for explicitly encouraging us to investigate the single-stack setting. The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under NHR project b185cb. NHR funding is provided by federal and Bavarian state authorities. NHR@FAU hardware is partially funded by the German Research Foundation (DFG), project 440719683. We also gratefully acknowledge the HPC resources provided by the JuliaV2 cluster at the University of Würzburg, funded by the German Research Foundation (DFG) under INST 93/1145-1 FUGG. The data science chair is part of the CAIDAS, the Center for Artificial Intelligence and Data Science, and is supported by the Bavarian High-Tech Agenda, which made this research possible.

References

- Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. [GQA: Training generalized multi-query transformer models from multi-head checkpoints](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.
- Jeremy Barnes, Robin Kurtz, Stephan Oepen, Lilja Øvrelid, and Erik Velldal. 2021. [Structured sentiment analysis as dependency graph parsing](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3387–3402, Online. Association for Computational Linguistics.
- Jeremy Barnes, Laura Oberlaender, Enrica Troiano, Andrey Kutuzov, Jan Buchmann, Rodrigo Agerri, Lilja Øvrelid, and Erik Velldal. 2022. [SemEval 2022 task 10: Structured sentiment analysis](#). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1280–1295, Seattle, United States. Association for Computational Linguistics.
- Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. [LLM2Vec: Large language models are secretly powerful text encoders](#). In *First Conference on Language Modeling*.
- Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Padó. 2014. GermEval 2014 named entity recognition shared task: companion paper. In *Workshop Proceedings of the 12th edition of the KONVENS conference*, pages 104–112.
- Wietse de Vries and Malvina Nissim. 2021. [As good as new. how to successfully recycle English GPT-2 to make models for other languages](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 836–846, Online. Association for Computational Linguistics.
- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. [Unified language model pre-training for natural language understanding and generation](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. [SemEval-2018 task 7: Semantic relation extraction and classification in scientific papers](#). In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 679–688, New Orleans, Louisiana. Association for Computational Linguistics.

- Kenji Imamura and Eiichiro Sumita. 2019. [Recycling a pre-trained BERT encoder for neural machine translation](#). In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 23–31, Hong Kong. Association for Computational Linguistics.
- Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah Smith. 2021. [Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation](#). In *International Conference on Learning Representations*.
- Roman Klinger and Philipp Cimiano. 2014. [The US-AGE review corpus for fine grained multi lingual opinion analysis](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 2211–2218, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. [Multilingual denoising pre-training for neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Meta AI. 2024. [Llama 3.2 model card](#).
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. [Abstractive text summarization using sequence-to-sequence RNNs and beyond](#). In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.
- Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. [Justifying recommendations using distantly-labeled reviews and fine-grained aspects](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.
- Jan Pfister and Andreas Hotho. 2024. [SuperGLEBer: German language understanding evaluation benchmark](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7904–7923, Mexico City, Mexico. Association for Computational Linguistics.
- Jan Pfister, Tom Völker, Anton Vlasjuk, and Andreas Hotho. 2025a. [BARTABSA++: Revisiting BARTABSA with decoder LLMs](#). In *Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025)*, pages 115–128, Vienna, Austria. Association for Computational Linguistics.
- Jan Pfister, Sebastian Wankerl, and Andreas Hotho. 2022. [SenPoi at SemEval-2022 task 10: Point me to your opinion, SenPoi](#). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1313–1323, Seattle, United States. Association for Computational Linguistics.
- Jan Pfister, Julia Wunderle, and Andreas Hotho. 2025b. [LLäMmleIn: Transparent, compact and competitive German-only language models from scratch](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2227–2246, Vienna, Austria. Association for Computational Linguistics.
- Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. [Leveraging pre-trained checkpoints for sequence generation tasks](#). *Transactions of the Association for Computational Linguistics*, 8:264–280.
- Mario Sängler, Ulf Leser, Steffen Kemmerer, Peter Adolphs, and Roman Klinger. 2016. [SCARE — the sentiment corpus of app reviews with fine-grained annotations in German](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 1114–1121, Portorož, Slovenia. European Language Resources Association (ELRA).
- Noam Shazeer. 2019. [Fast transformer decoding: One write-head is all you need](#). *Preprint*, arXiv:1911.02150.
- Sasha Spala, Nicholas Miller, Franck Dernoncourt, and Carl Dockhorn. 2020. [SemEval-2020 task 6: Definition extraction from free text with the DEFT corpus](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 336–345, Barcelona (online). International Committee for Computational Linguistics.
- Zhongxiang Sun, Zihua Si, Xiaoxue Zang, Kai Zheng, Yang Song, Xiao Zhang, and Jun Xu. 2025. [LargePiG for hallucination-free query generation: Your large language model is secretly a pointer generator](#). In *Proceedings of the ACM Web Conference 2025*, pages 4766–4779. Association for Computing Machinery.
- Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. [Multilingual translation with extensible multilingual pretraining and finetuning](#). *Preprint*, arXiv:2008.00401.

- Jörg Tiedemann, Mikko Aulamo, Daria Bakshandaeva, Michele Boggia, Stig-Arne Grönroos, Tommi Nieminen, Alessandro Raganato, Yves Scherrer, Raúl Vázquez, and Sami Virpioja. 2024. [Democratizing neural machine translation with OPUS-MT](#). *Language Resources and Evaluation*, 58(2):713–755.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. [Pointer networks](#). In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.
- Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. [Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.
- Michael Wojatzki, Eugen Ruppert, Sarah Holschneider, Torsten Zesch, and Chris Biemann. 2017. GermEval 2017: Shared Task on Aspect-based Sentiment in Social Media Customer Feedback. In *Proceedings of the GermEval 2017 – Shared Task on Aspect-based Sentiment in Social Media Customer Feedback*, pages 1–12, Berlin, Germany.
- Julia Wunderle, Anton Ehrmanntraut, Jan Pfister, Fotis Jannidis, and Andreas Hotho. 2026. [New encoders for German trained from scratch: Comparing ModernGBERT with converted LLM2Vec models](#). In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 10424–10446, Palma de Mallorca, Spain. ELRA Language Resource Association.
- Julia Wunderle, Jan Pfister, and Andreas Hotho. 2023. [Pointer networks: A unified approach to extracting German opinions](#). In *Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023)*, pages 127–138, Ingolstadt, Germany. Association for Computational Linguistics.
- Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. [Position-aware tagging for aspect sentiment triplet extraction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2339–2349, Online. Association for Computational Linguistics.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. [A unified generative framework for aspect-based sentiment analysis](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2416–2429, Online. Association for Computational Linguistics.
- Biao Zhang, Fedor Moiseev, Joshua Ainslie, Paul Suganthan, Min Ma, Surya Bhupatiraju, Fede Lebron, Orhan Firat, Armand Joulin, and Zhe Dong. 2025a. [Encoder-decoder Gemma: Improving the quality-efficiency trade-off via adaptation](#). *Preprint*, arXiv:2504.06225.
- Biao Zhang, Paul Suganthan, Gaël Liu, Ilya Philipov, Sahil Dua, Ben Hora, Kat Black, Gus Martins, Omar Sanseviero, Shreya Pathak, Cassidy Hardin, Francesco Visin, Jiageng Zhang, Kathleen Kenealy, Qin Yin, Xiaodan Song, Olivier Lacombe, Armand Joulin, Tris Warkentin, and Adam Roberts. 2025b. [T5Gemma 2: Seeing, reading, and understanding longer](#). *Preprint*, arXiv:2512.14856.
- Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. [Incorporating BERT into neural machine translation](#). In *International Conference on Learning Representations*.

A T5Gemma2 Architecture Overview

Referenced from section 3.3.

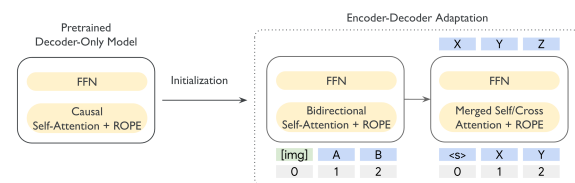


Figure 4: Overview of T5Gemma2. Reproduced from Zhang et al. (2025b): encoder and decoder parameters are initialized from a pretrained decoder-only model and then adapted with UL2. All word embeddings (in blue) are tied across encoder and decoder, and the decoder’s self- and cross-attention sub-layers are merged into a single block.

B Pointer Decoder Inputs

At each decoding step, the pointer head emits one index y_t into the joint set $[H^e; S]$ of eq. (1). The index denotes either an encoder position, which marks a span boundary, or a special token such as a sentiment label, grammar delimiter, or EOS. The

output grammar fixes the order of these choices: a sentiment closes each span tuple, and EOS closes the full sequence. The decoder itself consumes *token embeddings*, not pointer indices. Each emitted y_t must therefore be converted back into an input-token ID $\hat{y}_t$ before the next decoding step. A more complete treatment of the pointer-decoded Seq2Seq is given by Pfister et al. (2025a); we summarize only what a reader needs to interpret section 3.1.

Concretely, if y_t indexes the special-token block, $\hat{y}_t$ is that sentiment or boundary token; if y_t indexes an encoder position, $\hat{y}_t$ is the input subword token sitting at that position. From the decoder’s perspective each step therefore looks like ingesting a regular subword sequence, and the pointer mechanism stays confined to the output side. Special tokens (<ASP_BEGIN>, etc.) are added to the tokenizer and their embeddings initialized as the mean of their constituent sub-tokens. Both model parts use the encoder vocabulary; the decoder embedding matrix is resized to that vocabulary and adapted during fine-tuning, analogous to the vocabulary-replacement strategy of de Vries and Nissim (2021).

Training uses teacher forcing with cross-entropy on the pointer logits P_t .

C BARTABSA++ Enhancements We Inherit

The three modifications BARTABSA++ (Pfister et al., 2025a) makes to the inputs of eq. (1), referenced from section 3.1, are:

- (i) **feature normalization**, L2-normalizing H^e and S before the pointer score (turning it into a cosine similarity) and applying an RM-SNorm to the decoder output;
- (ii) **parametrized gating** of the encoder representations: the original BARTABSA blends the MLP-processed encoder output $\hat{H}^e$ with the raw input embeddings E^e through a fixed scalar α , $\bar{H}^e = \alpha\hat{H}^e + (1 - \alpha)E^e$; BARTABSA++ replaces α with a learned, input-dependent sigmoid gate (and adds an analogous decoder-side gate over H^d); and
- (iii) an **additional cross-attention block** placed between the decoder stack and the pointer head, querying the decoder hidden states H^d against the encoder representations $[H^e; S]$, whose projection weights are initialized from the pretrained final-layer cross-attention

when one exists (e.g. in BART, mBART, mT5).

We adopt all three throughout and do not analyze them further in this work.

D Hyperparameters

Unless stated otherwise, our pointer experiments use the BARTABSA++ (Pfister et al., 2025a) optimization setup: AdamW at a learning rate of 5×10^{-5} with weight decay 0.01, batch size 16, gradient clipping to norm 5.0, a polynomial decay learning-rate schedule, dropout 0.1, and bf16 mixed precision. These experiments train for up to 50 epochs and retain the checkpoint with the best development tuple F1. GNER instead uses one epoch, while summarization uses the development-selected four-epoch linear-schedule protocol described in section 4.3. Nothing is tuned per configuration. We run our experiments on L40 and L40S NVIDIA GPUs. Each takes between 30 minutes and 3 hours depending on the backbone size.

E Dataset Statistics

Table 4 lists split sizes and annotation counts for all datasets. SCARE, USAGE, 14lap, and 14res use three-way polarity; opener_en uses two-way polarity and additionally annotates opinion holders.

dataset	examples			annotations		
	train	dev	test	train	dev	test
SCARE	1100	140	133	1415	170	166
USAGE	1827	259	238	2970	405	395
opener_en	1744	249	499	2884	400	865
14lap	906	219	328	1460	346	543
14res	1266	310	492	2338	577	994
DEFT	6386	307	305	7322	337	358
SRE	284	66	152	982	246	367
GNER	24001	2199	5099	31543	2885	6692
CNN/DM	10000	1000	1000	10000	1000	1000

Table 4: Dataset sizes per split. Examples are sentences except for CNN/DM, where they are source articles. Annotations are sentiment triplets, DEFT/SRE relations, GNER entities, or CNN/DM reference summaries.

One example per dataset. To illustrate the shared triplet structure, one annotated example per dataset (word indices of respective span in parentheses, polarity in braces):

- **SCARE**, “Ohne dieses neue update kann ich gar nichts mehr machen!”: aspect *update* (3), opinion *gar nichts mehr machen* (6–9), {NEG}.

- **USAGE**, “*Super Waschmaschine*”: aspect *Waschmaschine* (1), opinion *Super* (0), {POS}.
- **opener_en**, “*They were there just to keep us happy.*”: aspect *They* (0), opinion *to keep ... happy* (4–5, 7), holder *us* (6), {POS}.
- **14lap**, “*Boot time is super fast, ...*”: aspect *Boot time* (0–1), opinion *fast* (4), {POS}.
- **14res**, “*The bread is top notch as well.*”: aspect *bread* (1), opinion *top notch* (3–4), {POS}.

opener_en carries the optional holder block of the grammar in section 3.1; the four other datasets do not annotate holders.

F Scaling Statistical Details

Table 5 reports the uncertainty intervals for the depth and scale effects summarized in section 5.2.

comparison	datasets	Δ F1	paired 95% CI
full encoder – e4, d4 fixed	both	+9.80	[8.47, 11.14]
d12 – d4, e4 fixed	both	+0.16	[−0.99, 1.30]
aligned e22 – e2	SCARE	+18.56	[15.73, 21.40]
aligned e22 – e2	USAGE	+17.49	[15.93, 19.05]
aligned d12 – d2	SCARE	+0.74	[−1.06, 2.53]
aligned d12 – d2	USAGE	+0.23	[−2.37, 2.84]
encoder-heavy – decoder-heavy, separate	both	+4.31	[−2.64, 11.26]
encoder-heavy – decoder-heavy, merged	both	+8.41	[4.35, 12.48]
1B encoder + 120M decoder – both 1B, merged	both	+7.05	[3.15, 10.94]

Table 5: Paired scaling contrasts across SCARE and USAGE. For the first two rows, “both” combines 20 matched differences: 2 datasets $\times$ 2 backbone pairs $\times$ 5 seeds. For the final three rows, it combines 10 matched differences: 2 datasets $\times$ 5 seeds. The aligned sweep averages each of 5 matched seeds over the other depth axis.

G English Transfer: Per-Dataset Numbers

Table 6 gives the per-dataset tuple F1 underlying section 5.5, with monolingual BART-base (Lewis et al., 2020) and multilingual mBART-large-50 (Tang et al., 2020; Pfister et al., 2022) as references. mBART is slightly above native BART-base on all three datasets, so it provides a multilingual calibration point on English as it does on German.

config (cross-attn)	14lap	14res	opener_en
<i>native BART</i>			
BART-base	57.61 $\pm$ 1.00	69.28 $\pm$ 1.04	51.71 $\pm$ 0.91
mBART-large-50	58.71 $\pm$ 1.68	71.60 $\pm$ 0.52	53.44 $\pm$ 0.57
<i>het. init</i>			
ModernBERT+LLaMA (separate)	45.98 $\pm$ 1.73	58.35 $\pm$ 1.60	39.57 $\pm$ 1.19
ModernBERT+LLaMA (merged)	45.55 $\pm$ 1.77	57.02 $\pm$ 1.98	38.92 $\pm$ 2.34
<i>hom. within-family init</i>			
LLaMA+LLaMA (separate)	49.05 $\pm$ 6.66	63.12 $\pm$ 4.62	44.65 $\pm$ 5.41
LLaMA+LLaMA (merged)	47.38 $\pm$ 6.42	66.14 $\pm$ 2.70	39.96 $\pm$ 11.23

Table 6: English transfer, 5 seeds per cell. Hom. and het. denote homogeneous and heterogeneous initialization. Values are test tuple-F1 mean $\pm$ std.

H Depth Transfer Across Tasks

Table 7 gives the complete structured-task depth results summarized in section 5.6.

task	pair	encoder-heavy	balanced	decoder-heavy
DEFT	hom.	45.23 $\pm$ 4.30	41.10 $\pm$ 3.88	39.19 $\pm$ 1.04
DEFT	het.	40.07 $\pm$ 0.93	35.73 $\pm$ 1.05	25.94 $\pm$ 3.73
SRE	hom.	8.41 $\pm$ 1.03	8.76 $\pm$ 0.82	3.37 $\pm$ 1.41
SRE	het.	4.37 $\pm$ 0.57	1.35 $\pm$ 0.90	0.42 $\pm$ 0.48
GNER	hom.	67.35 $\pm$ 2.57	54.49 $\pm$ 8.40	47.30 $\pm$ 3.71
GNER	het.	55.29 $\pm$ 0.36	50.74 $\pm$ 0.37	39.07 $\pm$ 0.98

Table 7: Merged-attention depth transfer, 3 seeds per cell. DEFT and SRE use e16/d4, e10/d10, and e4/d16; GNER uses e12/d4, e8/d8, and e4/d12. Values are test mean F1 $\pm$ std.

assembly	allocation	native	separate	merged	Δ
native BART	full	32.35 $\pm$ 0.28			
hom.	full		16.64 $\pm$ 1.13	24.04 $\pm$ 3.32	+7.40
het.	full		17.61 $\pm$ 1.90	15.06 $\pm$ 1.70	−2.54
hom.	e16/d4		16.80 $\pm$ 0.63	16.90 $\pm$ 0.62	+0.10
hom.	e10/d10		16.64 $\pm$ 0.05	20.32 $\pm$ 1.92	+3.68
hom.	e4/d16		16.71 $\pm$ 0.64	29.64 $\pm$ 0.48	+12.93

Table 8: CNN/DailyMail transfer and depth allocations, 3 seeds per cell. Values are standard ROUGE-1 $\times$ 100 mean $\pm$ std, and Δ is merged minus separate.

I Postmortem of Earlier Attention Experiments

An earlier version of this paper reported that separate cross-attention collapsed while merged attention trained, and used an ablation to attribute the difference to a fused residual update introduced by the merged attention mechanism. While reading and preparing the implementation for release, we found two faults that invalidated this interpretation: (i) the ModernGBERT decoder patch applied the causal mask to local-attention layers but not global-attention layers, so target positions could see future targets (only) during teacher-forced training; and (ii) later LLäMmleIn decoding steps rebuilt the pointer history in a way that replaced the configured start token with EOS. We corrected both paths and reran the attention comparison and ablation, and all current results in the paper use the corrected implementation. Interestingly, both unrelated faults distorted separate attention more severely than merged attention, producing the same apparently consistent pattern across the two attention mechanisms and hence the in hindsight incorrect story.

We retain the earlier ablation as a postmortem because the contrast may still suggest useful robustness questions, even though it cannot support the

original mechanism claim. R0 is the merged reference and Rref the separate reference. R1 through R6 were intended to isolate the six design differences between them by removing one part of merged attention at a time, until we arrived at Rref.

variant	change from R0	earlier F1	corrected F1
R0	merged reference	46.19 $\pm$ 2.69 (0/10)	44.30 $\pm$ 2.03 (0/10)
R1	split softmax	45.98 $\pm$ 3.98 (0/10)	45.33 $\pm$ 1.37 (0/10)
R2	fresh cross-Q	44.78 $\pm$ 3.75 (0/10)	43.21 $\pm$ 4.88 (0/10)
R3	no cross-Q RoPE	35.95 $\pm$ 12.20 (0/10)	43.41 $\pm$ 5.05 (0/10)
R4	fresh cross-O	43.47 $\pm$ 3.60 (0/10)	43.23 $\pm$ 2.50 (0/10)
R5	extra cross-attn norm	43.93 $\pm$ 3.55 (0/10)	41.43 $\pm$ 4.50 (0/10)
R6	split residual update	6.25 $\pm$ 6.91 (5/10)	42.29 $\pm$ 4.31 (0/10)
Rref	separate reference	5.94 $\pm$ 6.51 (5/10)	44.69 $\pm$ 1.83 (0/10)

Table 9: Test tuple-F1 mean $\pm$ std across SCARE and USAGE for the earlier faulty implementation and the corrected rerun. Each variant has 5 seeds per dataset and thus 10 runs; parentheses give the number below 5 F1.

The earlier matrix appeared to isolate the residual update: R6 and Rref fell near 6 F1, while R0 through R5 remained between 36 and 46 F1. After correction, for all 80 runs the means lie between 41.43 and 45.33 F1. R0 minus R6 is 2.01 F1 with a paired 95% CI of $[-1.49, 5.51]$. R0 minus Rref is -0.39 F1 with a CI of $[-2.13, 1.34]$.

The corrected ablation no longer supports the earlier mechanism claim, but it remains relevant to the paper’s core comparison of attention paths, and their asymmetry raises a separate question for future research: merged attention stayed usable under both faults while separate attention became brittle.

J Single-Stack Statistical Details

Tables 10 and 11 report the complete factorial results behind the headline rows in table 2.

topology	attention/visibility	query variant	params	SCARE F1	USAGE F1
two-stack	separate	enhanced	$\sim$ 232M	42.40 $\pm$ 3.07	44.57 $\pm$ 4.19
two-stack	merged	enhanced	203.8M	41.23 $\pm$ 2.55	42.83 $\pm$ 3.89
single-stack	causal	enhanced	103.7M	43.89 $\pm$ 0.91	43.01 $\pm$ 1.06
single-stack	causal	direct	102.5M	42.47 $\pm$ 2.12	43.34 $\pm$ 1.19
single-stack	prefix-visible	enhanced	103.7M	45.85 $\pm$ 1.80	46.05 $\pm$ 1.79
single-stack	prefix-visible	direct	102.5M	45.63 $\pm$ 2.19	46.44 $\pm$ 2.30
single-stack (1B)	causal	enhanced	1.06B	46.45 $\pm$ 3.34	50.41 $\pm$ 1.59
single-stack (1B)	causal	direct	1.05B	45.61 $\pm$ 2.27	50.10 $\pm$ 1.77
single-stack (1B)	prefix-visible	enhanced	1.06B	50.95 $\pm$ 1.82	53.95 $\pm$ 0.50
single-stack (1B)	prefix-visible	direct	1.05B	50.80 $\pm$ 2.16	54.32 $\pm$ 1.36

Table 10: Complete full-depth LLmMlein pointer-model comparison. The 120M rows use 10 matched seeds per cell and the 1B rows use 5 matched seeds per cell. Values are test tuple-F1 mean $\pm$ std.

scale	source visibility	query variant	SCARE F1	USAGE F1
134M	causal	direct	31.43 $\pm$ 2.63	27.50 $\pm$ 1.50
134M	causal	enhanced	32.35 $\pm$ 1.67	28.12 $\pm$ 2.24
134M	prefix-visible	direct	37.44 $\pm$ 2.99	36.68 $\pm$ 1.23
134M	prefix-visible	enhanced	38.67 $\pm$ 2.10	37.98 $\pm$ 2.89
1B	causal	direct	38.13 $\pm$ 2.50	40.39 $\pm$ 3.72
1B	causal	enhanced	39.84 $\pm$ 2.49	37.96 $\pm$ 3.27
1B	prefix-visible	direct	54.87 $\pm$ 2.60	57.11 $\pm$ 0.89
1B	prefix-visible	enhanced	57.05 $\pm$ 1.64	56.82 $\pm$ 1.10

Table 11: Complete full-depth ModernGBERT single-stack matrix. The 134M rows use 10 seeds per cell and the 1B rows use 5. Values are test tuple-F1 mean $\pm$ std.

comparison	scale / condition	Δ F1	paired 95% CI
prefix-visible — causal	134M	+7.85	[6.58, 9.11]
prefix-visible — causal	1B	+17.38	[16.17, 18.59]
enhanced — direct	134M	+1.02	[−0.01, 2.05]
enhanced — direct	1B	+0.29	[−1.47, 2.05]
1B — 134M	causal, direct	+10.27	[7.75, 12.80]
1B — 134M	causal, enhanced	+8.80	[7.19, 10.40]
1B — 134M	prefix, direct	+19.32	[16.30, 22.34]
1B — 134M	prefix, enhanced	+18.29	[15.16, 21.42]

Table 12: Paired t -test results for ModernGBERT single-stack factor effects. Visibility effects average each seed over both datasets and query variants; refinement effects average over both datasets and visibility patterns. Scale effects average over both datasets within each visibility/refinement condition. The 134M visibility and refinement contrasts use 10 seeds, while the 1B and scale contrasts use 5.

topology	dataset	GPU h/run	greedy ms/example	train GiB	test GiB
single-stack	SCARE	0.17	26.63	1.52	1.43
two-stack	SCARE	0.28	44.46	2.44	2.89
single-stack	USAGE	0.34	45.60	3.71	1.44
two-stack	USAGE	0.55	82.03	4.91	2.89

Table 13: Resource use for the 120M prefix-visible enhanced single stack and the merged same-family two-stack model. All runs used one NVIDIA L40. GPU-hours are means per run, greedy latency is the median per example, and memory columns report the maximum peak allocation across seeds.

The single stack lowers mean training GPU-hours by 41% on SCARE and 39% on USAGE and uses about half the peak test memory.

Table 14 gives the paired intervals underlying the two-stack, source-visibility, refinement, and scale comparisons in section 5.3. We use paired t -tests over the seed-level F1 differences. All rows report two-sided 95% intervals against zero.

comparison	dataset	Δ F1	interval
causal enhanced — merged	SCARE	+2.66	[1.31, 4.01]
causal enhanced — merged	USAGE	+0.18	[−2.59, 2.94]
causal enhanced — separate	SCARE	+1.49	[−0.85, 3.84]
causal enhanced — separate	USAGE	−1.56	[−4.58, 1.45]
prefix enhanced — merged	SCARE	+4.62	[1.82, 7.42]
prefix enhanced — merged	USAGE	+3.21	[0.68, 5.75]
prefix enhanced — separate	SCARE	+3.45	[0.55, 6.35]
prefix enhanced — separate	USAGE	+1.47	[−1.79, 4.74]
prefix — causal, enhanced	SCARE	+1.96	[0.26, 3.66]
prefix — causal, enhanced	USAGE	+3.04	[1.59, 4.49]
prefix — causal, direct	SCARE	+3.16	[1.69, 4.62]
prefix — causal, direct	USAGE	+3.09	[0.88, 5.30]
enhanced — direct, prefix	SCARE	+0.22	[−1.80, 2.24]
enhanced — direct, prefix	USAGE	−0.39	[−2.43, 1.66]
visibility $\times$ refinement	SCARE	−1.20	[−2.82, 0.42]
visibility $\times$ refinement	USAGE	−0.05	[−3.03, 2.93]
1B causal enhanced — 120M	SCARE	+2.58	[−1.63, 6.79]
1B causal enhanced — 120M	USAGE	+7.06	[5.38, 8.74]
1B prefix-visible — causal, direct	SCARE	+5.18	[1.47, 8.90]
1B prefix-visible — causal, direct	USAGE	+4.21	[2.80, 5.63]
1B prefix-visible — causal, enhanced	SCARE	+4.50	[−1.28, 10.28]
1B prefix-visible — causal, enhanced	USAGE	+3.54	[1.82, 5.26]
1B enhanced — direct, prefix-visible	SCARE	+0.15	[−3.69, 4.00]
1B enhanced — direct, prefix-visible	USAGE	−0.37	[−2.27, 1.53]
1B single — encoder-heavy merged	SCARE	−2.85	[−10.49, 4.80]
1B single — encoder-heavy merged	USAGE	−0.32	[−3.07, 2.43]
1B single — encoder-heavy separate	SCARE	+2.61	[−7.16, 12.38]
1B single — encoder-heavy separate	USAGE	+5.74	[−0.49, 11.97]
1B single — decoder-heavy merged	SCARE	+5.04	[−2.05, 12.13]
1B single — decoder-heavy merged	USAGE	+8.62	[−0.06, 17.30]
1B single — all-1B merged	SCARE	+2.85	[−7.29, 12.98]
1B single — all-1B merged	USAGE	+8.08	[1.78, 14.39]

Table 14: Paired t -test comparisons for the LLäMmleIn single-stack experiments. The 120M contrasts use 10 matched seeds; the 1B contrasts use 5. Encoder-heavy pairs a 1B encoder with a 120M decoder, decoder-heavy reverses that allocation, and all-1B uses both 1B stacks. All intervals are two-sided 95% CIs over matched seed differences.