Cross-Decoder Retargeting of Soft Compressor for Long-Context Compression

Anonymous ACL submission

Abstract

Soft compression stores a text span as k continuous compressed vectors for a frozen decoder, but existing compressors are usually tied to the decoder used during training. We study whether those vectors can be retargeted to a different frozen decoder through its input-embedding interface. Our benchmark covers seven target decoders, three vector budgets, and directed source–target pairs, comparing **Random**, shared-embedding least squares (**LS**), **Converter-only**, **Enc-Conv**, and target-native ORIGIN. Transfer is feasible but decoder-conditioned: trainable retargeting can approach ORIGIN BLEU on decoder-compatible targets, while LS is often close to Random despite using shared token anchors. Perturbation and geometry probes show that decoders expose different readable regions for continuous inputs. Finally, pair-level readability scores provide moderate pre-transfer rankings for choosing sources, targets, and whether end-to-end retargeting is worth its cost.[1]

1 Introduction

Long-context systems increasingly use soft compressors: a text segment is replaced by k continuous compressed vectors that a frozen decoder reads through its input-embedding interface. This avoids attending to every original token and can preserve information beyond discrete pruning or summarization. The practical weakness is portability. A compressor trained for one decoder may fail when the deployment decoder changes, when a model is upgraded, or when a service must support several decoder families.

This paper studies decoder-interface portability: whether compressed vectors produced for one decoder can be retargeted to another frozen decoder. Our claim is that soft-compressed vectors are decoder-conditioned continuous inputs, not decoder-agnostic semantic codes. A source compressor may preserve the segment for its own decoder, yet remain unreadable to another decoder if its vectors fall outside the target's usable region. We use autoregressive reconstruction as a controlled probe of this interface.

We organize the study around three questions. **RQ1: Can compressed vectors be transferred at all?** Section 5 compares Random, LS alignment in shared token embeddings, Converter-only, Enc-Conv, target-native ORIGIN, and a matched-budget from-scratch control. **RQ2: Why does transfer succeed or fail?** Section 6 tests whether failures are consistent with source vectors missing the target decoder's readable region. **RQ3: Can pre-transfer diagnostics guide retargeting choices?** Section 7 evaluates source–target readability scores for ranking candidate sources, targets, and methods before training.

The answer is qualified. Compressed vectors are reusable, but not as portable semantic summaries. LS can underperform Random, while trainable retargeting recovers much more of the target-native reconstruction quality. Transfer is also directional: a source distribution can be readable after retargeting into one decoder while the reverse direction remains difficult. This argues against a universal latent-space view and points to source-conditioned compressed-vector manifolds interacting with target-readable regions.

Gaussian perturbation and native-geometry statistics support this view: target-native compressed vectors occupy finite readable neighborhoods that differ across decoder families. Pair-specific readable-geometry scores then provide moderate pre-transfer guidance for ranking source–target pairs and method choices, while reconstruction remains a decoder-interface proxy rather than a claim about all downstream tasks.

Our main contributions are:

[1] Code and result artifacts are available at https://anonymous.4open.science/r/SC_Retargeting-8A27/.

- We formulate compressed-vector retargeting as a decoder-interface portability problem for soft compression and build a controlled benchmark across decoder families, transfer directions, and vector budgets.

- We show that compressed vectors are transferable but not decoder-agnostic: trainable retargeting is effective, while transfer remains target-dependent and directional.

- We support a target-readable-region explanation with perturbation and geometry diagnostics, and show that pair-specific compatibility scores provide lightweight pre-transfer guidance.

2 Related Work

2.1 Long Context Compression

Context compression reduces long-context cost by pruning, summarizing, rewriting, or compressing inputs (Jiang et al., 2023b, 2024; Pan et al., 2024). Continuous compression instead encodes context into learned vectors injected into the decoder embedding interface (Chevalier et al., 2023; Ge et al., 2024; Wang et al., 2024). PCC (Dai et al., 2025) and ARC (Pilchen et al., 2025) adapt compressors through continued training, but direct cross-decoder portability of compressed vectors remains less studied.

2.2 Capacity and Limits of Soft Compression

Soft compression can store substantial information in few vectors (Kuratov et al., 2025; Mezentsev and Oseledets, 2025). Our question is not whether these vectors have capacity, but whether the capacity remains usable after changing the decoder that reads them.

2.3 Transferability of Continuous Representations

Soft-prompt transfer (Su et al., 2022), cross-model embedding conversion (Yoon and Arik, 2025), universal embedding geometry (Jha et al., 2025), and modality alignment (Norelli et al., 2024; Wu et al., 2025) all study portable continuous representations. They are not drop-in baselines here: prompt transfer moves task vectors while the original text remains present, and embedding transfer aligns token or sentence spaces rather than compressor-produced decoder inputs. We therefore compare methods that operate on the same compressed-vector interface: Random, LS, Converter-only, Enc-Conv, and target-native ORIGIN.

We also test the anchor idea from zero-shot continuous prompt transfer (Wu et al., 2024). It works in a closer prompt-transfer setting, but performs poorly for input-dependent compressed segments with heterogeneous tokenizers (Appendix C). Coordinate-search variants backpropagate through the frozen target for each vector refinement, which is costly relative to lightweight retargeting or inference-time compression.

3 Method

This section formalizes compressed-vector retargeting as decoder-interface portability. Figure 1 shows the pipeline. First, a source compressor encodes a sampled text segment into k continuous compressed vectors for a source decoder. Second, a retargeting method maps or regenerates these vectors for a target decoder space. Third, the frozen target decoder receives the retargeted vectors through its input-embedding interface and reconstructs the same segment.

3.1 Problem Definition

Let $\mathcal{X}$ denote the text-segment distribution and let $x \sim \mathcal{X}$ be a segment to compress. A soft compressor paired with a source decoder D_s produces a sequence of k continuous vectors:

$$Z_s = \text{Compressor}_s(x) \in \mathbb{R}^{k \times d_s}, \qquad (1)$$

where d_s is the input-embedding dimension of D_s. The retargeting problem is to construct a target-side continuous input

$$Z_t = R_{s \to t}(x, Z_s) \in \mathbb{R}^{k \times d_t}, \qquad (2)$$

for a frozen target decoder D_t with embedding dimension d_t. Depending on the method, $R_{s \to t}$ may only transform the already-computed Z_s or may also regenerate compressed vectors from x.

The objective is not merely to move a representation in vector space. The target decoder consumes Z_t as a prefix-like continuous input and generates a reconstruction $\hat{x}_t$. We call retargeting successful when, for typical $x \sim \mathcal{X}$, $\hat{x}_t$ preserves the content of x and approaches the reconstruction quality of a target-native compressor. In experiments this is measured by Acc, BLEU, and BLEU retention relative to the target-native ORIGIN reference.

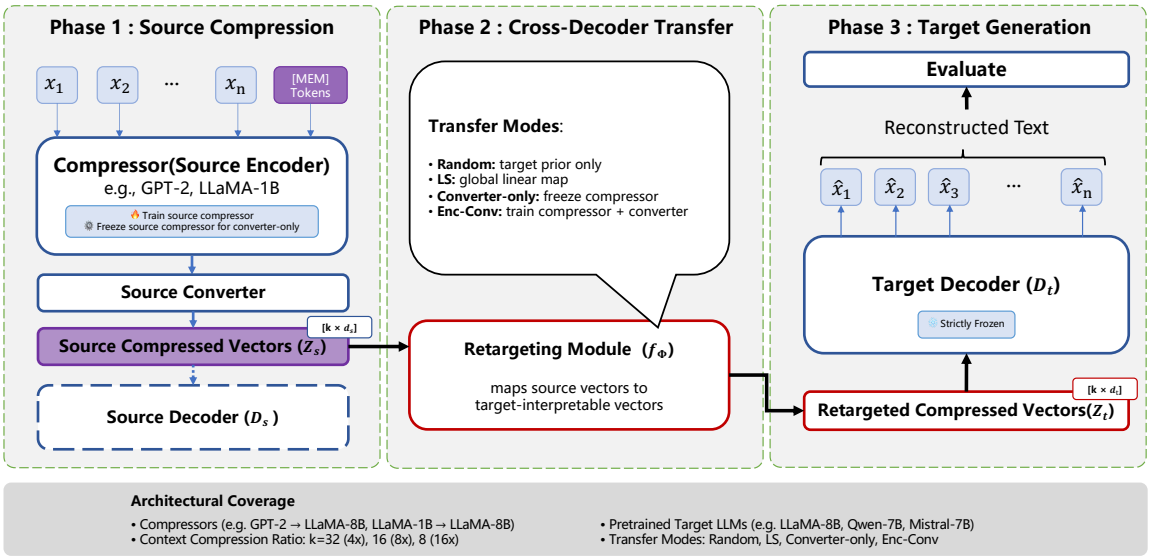

Figure 1: Overview of compressed-vector retargeting. A source compressor produces continuous compressed vectors for a source decoder. Retargeting methods transform or regenerate these vectors so that a frozen target decoder can consume them. The target-native compressor provides the upper reference.

3.2 Soft Compressor Design and Training

The compressor uses a two-module design to encode input segment x into a fixed-length sequence of compressed vectors. Text reconstruction is used as a controlled decoder-interface proxy: it tests whether information carried by continuous inputs can be read by a frozen target decoder across many source–target pairs. During training, the decoder being conditioned remains frozen, requiring the compressor or retargeting module to produce vectors that can be interpreted through the decoder input interface. The model is optimized with teacher forcing, and the loss is computed only on target token positions so that compressed-vector positions do not directly contribute to the objective.

3.3 Retargeting Strategies

To examine the cross-model transfer landscape, four retargeting methods and one target-native reference are compared:

- **Random (Lower Bound):** Random vectors with the same shape as the target compressed vectors are used as a lower reference. This baseline measures how much reconstruction can be achieved from the target decoder prior alone.
- **Least Squares (LS) Alignment:** A global linear map is estimated from shared token embeddings between the source and target decoders and then applied to the source compressed vectors. This method tests whether retargeting can be approximated by a single embedding-space alignment.
- **Converter-Only:** The source compressor is frozen, and a trainable converter maps its compressed vectors into the target decoder space, $Z_t = f_\phi(Z_s)$. This setting evaluates whether a post-hoc nonlinear transformation is sufficient when the original compressed-vector generator remains fixed.
- **Encoder-Converter (End-to-End):** The compressor encoder and the converter are jointly optimized while the target decoder remains frozen, $Z_t = f_\phi(\text{Compressor}_\theta(x))$. This method allows the compressed-vector generator itself to adapt to the target decoder, rather than only transforming vectors produced for the source decoder.
- **Target-native (Upper Reference):** A compressor is trained directly for the target decoder. This setting bypasses cross-model transfer and serves as the ORIGIN reference for quantifying the performance loss introduced by retargeting.

Appendix A reports decoder and training settings, and Appendix B gives LS, anchor, and geometry-prior implementation details.

4 Experimental Setup

This section describes the controlled setup for evaluating compressed-vector retargeting. Method details are given in Section 3; here, the focus is on models, data, training protocol, and metrics.

4.1 Decoders and Transfer Pairs

Experiments use seven autoregressive decoders from the LLaMA (Grattafiori et al., 2024), Qwen (Yang et al., 2024), and Mistral (Jiang et al., 2023a) families, covering 1B–8B parameter scales and different tokenizer conventions. Source–target pairs are constructed to include within-family transfer, cross-family transfer, scale changes, and both transfer directions.

4.2 Data

All experiments use text segments sampled from the FineWeb-10BT corpus (Penedo et al., 2024). The training split is used for target-native compressors and trainable retargeting modules, while the held-out split is used for evaluation. Each segment is truncated to 128 tokens under the corresponding tokenizer, controlling for sequence-length differences across decoders.

4.3 Training and Evaluation Protocol

All decoders are frozen. Target-native compression trains only the compressor for the target decoder. Converter-only retargeting freezes the source compressor and trains only the converter, whereas encoder–converter retargeting jointly trains the compressed-vector generator and converter. Random and LS require only vector construction and no end-to-end optimization.

Experiments evaluate 8, 16, and 32 compressed vectors. Main results use 32 vectors; shorter settings are used to analyze compression-budget effects. Trainable components use the reconstruction objective in Section 3. Training uses bf16 precision, one epoch, and a learning rate of 10^{-4}. Evaluation uses 1,000 held-out examples. Autoregressive reconstruction uses greedy decoding with at most 128 generated tokens.

4.4 Metrics

Two metrics are reported in the tables. **Acc** is token-level accuracy on the target vocabulary under gold-prefix conditioning. **BLEU** (Papineni et al., 2002) measures autoregressive reconstruction quality when the decoder receives only compressed vectors. We also compute sentence-level ROUGE-L (Lin, 2004); it tracks BLEU closely here, so tables omit it and the evaluation bundle includes the values. For transfer methods, BLEU is compared with the target-native ORIGIN result:

$$\Delta_{\text{BLEU}} = 1 - \frac{\text{BLEU}_{\text{transfer}}}{\text{BLEU}_{\text{Origin}}} \tag{3}$$

Lower relative gaps indicate that retargeting approaches target-native compression quality. Maintable subscripts report Δ_{BLEU} in percent.

5 Transferability of Compressed Vectors

This section examines whether compressed vectors learned for one decoder can be retargeted to another decoder. We focus on the 32-vector setting because target-native compression is already strong and therefore provides a strict reference. Table 1 reports the main results with the LLaMA-1B encoder backbone, while the GPT-2-backbone counterpart is provided in Appendix E. Figure 2 summarizes best-transfer BLEU retention across directed source–target pairs, averaging over both encoder backbones. Retention normalizes transfer BLEU by the row-specific target-native reference:

$$\text{Retention} = \frac{\text{BLEU}_{\text{METHOD}}}{\text{BLEU}_{\text{ORIGIN,TARGET}}}. \tag{4}$$

Throughout this section, table subscripts denote relative BLEU change against the same target-native ORIGIN baseline.

5.1 Transfer Is Possible, but Not by Global Alignment

Table 1 shows that compressed vectors are not locked to a single decoder: trainable retargeting can recover a large fraction of target-native reconstruction quality. The important qualification is how this recovery happens. Random vectors provide a lower reference, and LS alignment is often no better, sometimes worse, despite using shared token embeddings. This rules out a simple view in which compressed-vector spaces can be matched by a global embedding map. Converter-only and Enc-Conv succeed because they learn a decoder-facing transformation of the compressed-vector sequence; Enc-Conv is strongest in most rows because it can also change how the vectors are generated.

This also motivates keeping the main table to interface-matched baselines: anchor prompt transfer may work for short prompt vectors

Block	Transfer Pair	Random	LS	Converter-only	Enc-Conv	Origin
A → A	LLaMA-1B → Qwen-1.5B	0.404/0.127$_{-87\%}$	0.338/0.093$_{-90\%}$	0.802/0.341$_{-65\%}$	**0.969/0.822**$_{-16\%}$	0.995/0.978
	Qwen-1.5B → LLaMA-1B	0.354/0.092$_{-91\%}$	0.005/0.000$_{-100\%}$	0.967/0.871$_{-13\%}$	**0.991/0.951**$_{-4\%}$	1.000/0.996
A → B	LLaMA-1B → Qwen-3B	0.323/0.093$_{-90\%}$	0.409/0.137$_{-86\%}$	0.453/0.027$_{-97\%}$	**0.765/0.306**$_{-68\%}$	0.994/0.971
	LLaMA-1B → LLaMA-3B	0.369/0.100$_{-90\%}$	0.239/0.030$_{-97\%}$	0.997/0.983$_{-1\%}$	**0.997/0.990**$_{-1\%}$	1.000/0.997
	Qwen-1.5B → Qwen-3B	0.323/0.093$_{-90\%}$	0.423/0.145$_{-85\%}$	0.785/0.317$_{-67\%}$	**0.923/0.701**$_{-28\%}$	0.994/0.971
	Qwen-1.5B → LLaMA-3B	0.369/0.100$_{-90\%}$	0.012/0.000$_{-100\%}$	0.976/0.933$_{-6\%}$	**0.996/0.977**$_{-2\%}$	1.000/0.997
A → C	LLaMA-1B → LLaMA-8B	0.409/0.115$_{-89\%}$	0.214/0.033$_{-97\%}$	**0.998/0.994**$_{-0\%}$	0.999/0.993$_{-0\%}$	1.000/0.996
	LLaMA-1B → Mistral-7B	0.423/0.103$_{-89\%}$	0.157/0.009$_{-99\%}$	0.987/0.939$_{-4\%}$	**0.995/0.972**$_{-1\%}$	0.996/0.981
	LLaMA-1B → Qwen-7B	0.432/0.164$_{-83\%}$	0.430/0.160$_{-83\%}$	0.973/0.897$_{-7\%}$	**0.988/0.943**$_{-2\%}$	0.996/0.961
	Qwen-1.5B → LLaMA-8B	0.409/0.115$_{-89\%}$	0.001/0.000$_{-100\%}$	0.988/0.952$_{-4\%}$	**0.997/0.978**$_{-2\%}$	1.000/0.996
	Qwen-1.5B → Mistral-7B	0.423/0.103$_{-89\%}$	0.037/0.000$_{-100\%}$	0.981/0.914$_{-7\%}$	**0.992/0.962**$_{-2\%}$	0.996/0.981
	Qwen-1.5B → Qwen-7B	0.432/0.129$_{-87\%}$	0.430/0.124$_{-87\%}$	0.971/0.885$_{-8\%}$	**0.986/0.925**$_{-4\%}$	0.996/0.961
B → B	Qwen-3B → LLaMA-3B	0.369/0.100$_{-90\%}$	0.051/0.000$_{-100\%}$	0.972/0.902$_{-10\%}$	**0.995/0.974**$_{-2\%}$	1.000/0.997
	LLaMA-3B → Qwen-3B	0.323/0.093$_{-90\%}$	0.423/0.147$_{-85\%}$	0.655/0.161$_{-83\%}$	**0.934/0.742**$_{-24\%}$	0.994/0.971
B → C	Qwen-3B → LLaMA-8B	0.409/0.128$_{-87\%}$	0.030/0.000$_{-100\%}$	0.986/0.939$_{-6\%}$	**0.997/0.984**$_{-1\%}$	1.000/0.996
	Qwen-3B → Mistral-7B	0.423/0.103$_{-89\%}$	0.022/0.000$_{-100\%}$	0.979/0.906$_{-8\%}$	**0.991/0.945**$_{-4\%}$	0.996/0.981
	Qwen-3B → Qwen-7B	0.432/0.164$_{-83\%}$	0.408/0.128$_{-87\%}$	0.966/0.846$_{-12\%}$	**0.982/0.913**$_{-5\%}$	0.996/0.961
	LLaMA-3B → LLaMA-8B	0.409/0.128$_{-87\%}$	0.108/0.004$_{-100\%}$	0.998/0.989$_{-1\%}$	**0.998/0.994**$_{-0\%}$	1.000/0.996
	LLaMA-3B → Mistral-7B	0.423/0.103$_{-89\%}$	0.179/0.007$_{-99\%}$	0.986/0.934$_{-5\%}$	**0.994/0.978**$_{-0\%}$	0.996/0.981
	LLaMA-3B → Qwen-7B	0.432/0.164$_{-83\%}$	0.437/0.161$_{-83\%}$	0.963/0.857$_{-11\%}$	**0.987/0.945**$_{-2\%}$	0.996/0.961
C → C	LLaMA-8B → Mistral-7B	0.423/0.103$_{-89\%}$	0.080/0.000$_{-100\%}$	0.978/0.892$_{-9\%}$	**0.984/0.921**$_{-6\%}$	0.996/0.981
	LLaMA-8B → Qwen-7B	0.432/0.164$_{-83\%}$	0.436/0.152$_{-84\%}$	0.952/0.824$_{-14\%}$	**0.985/0.936**$_{-3\%}$	0.996/0.961
	Mistral-7B → LLaMA-8B	0.409/0.128$_{-87\%}$	0.050/0.000$_{-100\%}$	0.923/0.778$_{-22\%}$	**0.975/0.873**$_{-12\%}$	1.000/0.996
	Mistral-7B → Qwen-7B	0.432/0.164$_{-83\%}$	0.264/0.039$_{-96\%}$	0.885/0.678$_{-30\%}$	**0.974/0.894**$_{-7\%}$	0.996/0.961
	Qwen-7B → LLaMA-8B	0.409/0.128$_{-87\%}$	0.016/0.000$_{-100\%}$	0.991/0.962$_{-3\%}$	**0.998/0.985**$_{-1\%}$	1.000/0.996
	Qwen-7B → Mistral-7B	0.423/0.103$_{-89\%}$	0.093/0.001$_{-100\%}$	0.983/0.923$_{-6\%}$	**0.992/0.940**$_{-4\%}$	0.996/0.981

Table 1: Main transfer results at k=32 with a LLaMA-1B encoder backbone; pair columns read source native decoder → target decoder. Blocks group decoder scales: A = LLaMA-1B and Qwen-1.5B; B = LLaMA-3B and Qwen-3B; C = LLaMA-8B, Mistral-7B, and Qwen-7B. Each cell reports Acc/BLEU; transfer subscripts report Δ_{BLEU} in percent against row-specific target-native ORIGIN. Cell background runs from weaker (red) to stronger (green) transfer BLEU within each row. Bold and underline mark the best and worst transfer BLEU in each row.

and close model pairs (Wu et al., 2024), but our input-dependent compressed segments face sparse tokenizer anchors and a mismatch between compressed-vector and token-embedding spaces.

Appendix C shows Anchor-BP/Procrustes remain far below trainable retargeting; existing embedding-transfer and prompt-tuning methods (Su et al., 2022; Yoon and Arik, 2025; Jha et al., 2025) are therefore related references rather than direct baselines for decoder-interface compressed-vector retargeting.

5.2 The Target Decoder Determines Readability

The retention heatmap in Figure 2 gives the main pattern more clearly than any single row. Reading by columns, LLaMA, Mistral, and Qwen-7B targets generally retain a large fraction of their ORIGIN BLEU, while incoming transfer to smaller Qwen targets is more variable. This column-wise structure means that the same source-side information can be readable for one target decoder but poorly matched to another. Since every cell is normalized by the target-native ORIGIN result, this is not simply a statement about native compression quality or absolute BLEU scale. The target decoder determines whether the retargeted continuous inputs fall in a region it can read.

This target-readable view also explains the directional structure. A source–target pair that transfers well in one direction can remain difficult in the reverse direction. The Qwen rows and columns illustrate the distinction: vectors trained for a Qwen decoder can often be retargeted to LLaMA or Mistral targets, while the reverse direction depends much more on the specific Qwen target. Such asymmetry is inconsistent with symmetric space alignment. It suggests that each compressor learns a decoder-conditioned distribution, and transfer succeeds only when that distribution can be reshaped into the target's readable interface.

5.3 Vector Budget Changes Capacity, Not the Constraint

Increasing the compressed-vector budget improves target-native reconstruction and generally helps

Target	Scratch	Full Origin	Enc-Conv	Scratch/Orig.
LLaMA-1B	0.281	0.996	0.996	28.3%
LLaMA-3B	0.848	0.997	0.990	85.0%
LLaMA-8B	0.024	0.996	0.993	2.5%
Qwen-1.5B	0.029	0.978	0.822	3.0%
Qwen-3B	0.013	0.971	0.306	1.4%
Qwen-7B	0.021	0.961	0.943	2.2%
Mistral-7B	0.829	0.981	0.972	84.6%

Table 2: Matched-budget target-native training at k=32. **Scratch** trains a randomly initialized target-native compressor for one epoch on the same 100k segments used by transfer. **Full Origin** is the target-native compressor trained on the full 1M-segment pool. **Enc-Conv** retargets from the LLaMA-1B source with the same 100k transfer budget. All entries are BLEU.

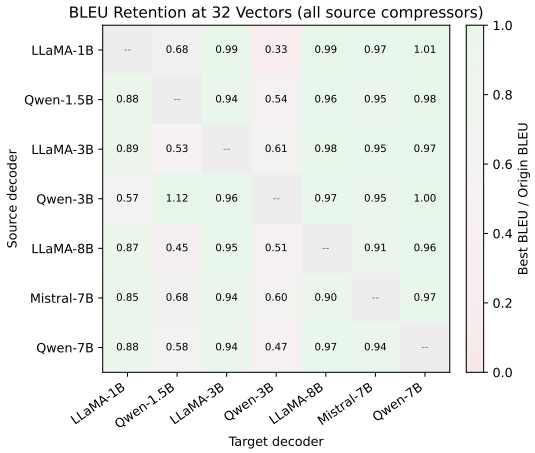

Figure 2: Target-dependent transferability at k=32, averaged over GPT-2 and LLaMA-1B encoder backbones. Each cell reports best-transfer BLEU retention: the best BLEU among Random, LS, Converter-only, and Enc-Conv divided by the corresponding target-native ORIGIN BLEU. Values above 1.0 mean transfer BLEU exceeds that ORIGIN reference.

trainable retargeting, but it does not remove the target-compatibility constraint. The same qualitative pattern holds across 8, 16, and 32 vectors: trainable retargeting is stronger than random or linear baselines, target-side effects remain prominent, and direction matters. Detailed budget results are deferred to Appendix E.

5.4 Matched-Budget Scratch Training

The main ORIGIN reference uses the full target-native training pool, whereas retargeting uses only $0.1\times$ that data. Table 2 therefore adds a matched-budget target-native compressor trained from random initialization on the same 100k segments used by transfer. The result separates retargeting from simply spending a small target budget. On several targets, especially Qwen targets and LLaMA-8B, matched-budget Scratch is far below Enc-Conv. This indicates that Enc-Conv is not merely learning a new target compressor from scratch; it benefits from reusing the source compressor. LLaMA-3B and Mistral-7B are easier exceptions where Scratch is competitive, showing that the advantage of retargeting is target-dependent rather than universal.

Overall, RQ1 shows that compressed vectors carry reusable information, but not as decoder-agnostic semantic summaries. Their portability is governed by whether a target decoder can read the transformed compressed-vector sequence through its own continuous input interface.

6 Mechanisms of Compressed-Vector Retargeting

Transfer can fail even when source information is preserved, if the transformed compressed-vector sequence is unreadable to the target decoder. This section tests a target-readable-region hypothesis. A compressed vector is a continuous decoder input, so its usefulness is defined by the decoder that consumes it. Retargeting must therefore preserve source content while placing the resulting vectors near a target-readable compressed-vector manifold.

6.1 Readable Regions Have Measurable Width

We first perturb target-native compressed vectors with Gaussian noise. For each unperturbed vector row z_i, we add i.i.d. noise with standard deviation $\sigma\|z_i\|_2$ and measure reconstruction accuracy. Accuracy is normalized by the unperturbed run for the same compressor. The half-decay threshold σ_{50} is the value of σ at which this relative accuracy reaches 0.5, linearly interpolated between neighboring grid points when necessary. Smaller σ_{50} means that reconstruction breaks under smaller vector perturbations, which we interpret as a narrower local readable neighborhood under this probe.

Figure 3 and Table 3 show that the threshold differs across native compressors, indicating that reconstruction depends on compressed-vector geometry and that readable neighborhoods vary across decoders.

6.2 Readable Regions Also Have Distinct Geometry

Perturbation measures local width; native compressed-vector statistics describe the shape

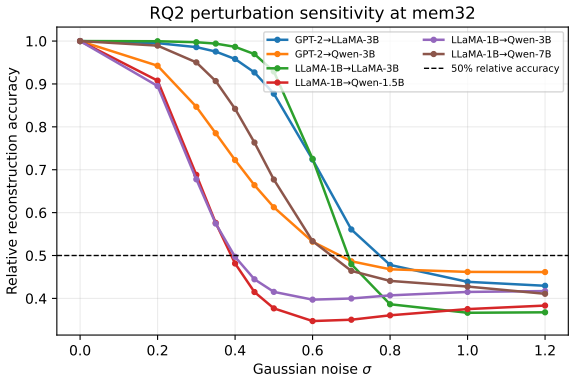

Figure 3: Gaussian perturbation of target-native compressed vectors at k=32. **Acc** is normalized by the unperturbed native run (Section 4.4). Lower σ_{50} indicates that reconstruction accuracy falls to half of its reference value under smaller vector perturbations.

Native compressor	d	σ_{50}
GPT-2→LLaMA-3B	3072	0.773
LLaMA-1B→LLaMA-3B	3072	0.692
LLaMA-1B→LLaMA-1B	2048	0.618
LLaMA-1B→Qwen-7B	3584	0.648
GPT-2→Qwen-3B	2048	0.671
LLaMA-1B→Qwen-3B	2048	0.398
LLaMA-1B→Qwen-1.5B	1536	0.390

Table 3: Gaussian half-decay thresholds for native compressors at k=32. d is the target decoder hidden size. σ_{50} is the relative noise level $\sigma\|z\|_2$ at which accuracy (Section 4.4) falls to half of the unperturbed native run, with linear interpolation between grid points. Lower σ_{50} means narrower local tolerance to compressed-vector noise.

of the region. Let $X \in \mathbb{R}^{N\times d}$ stack all k vector rows across evaluated examples for a native compressor, and let $\tilde{X}$ subtract the column mean. With sample covariance $\Sigma = (N-1)^{-1}\tilde{X}^\top\tilde{X}$ and eigenvalues $\lambda_j \geq 0$, we define effective rank as the participation ratio

$$r_{\text{eff}} = \frac{(\sum_j \lambda_j)^2}{\sum_j \lambda_j^2}. \tag{5}$$

We also report the mean pairwise cosine among the k compressed vectors within each segment, averaged over segments, and the mean row norm of X. Statistics are computed on the same 1,000 held-out segments used elsewhere, based on native compressor outputs at the decoder block interface (i.e., the actual d-dimensional inputs consumed under native compression), rather than the static token-embedding matrix alone. Table 4 summarizes the family-level contrast. At 32 vectors, Qwen-family

Family	Eff. rank	Cosine	Norm
LLaMA	18.41	0.552	41.94
Qwen	3.85	0.778	180.38

Table 4: Native compressed-vector geometry at k=32, averaged within LLaMA and Qwen decoder families over 128 held-out segments per native compressor (Mistral has a single target here and is omitted). Lower effective rank and higher within-segment cosine indicate a more concentrated native vector distribution. **Norm** is the mean ℓ_2 norm of compressed-vector rows.

native compressed vectors are more concentrated than the LLaMA-family counterparts in the analyzed runs: average effective rank is lower, within-segment vector cosine is higher, and row norms are larger. These differences do not imply that every Qwen transfer is difficult, but they show that target decoders can use distinct compressed-vector geometries. A retargeter must therefore match not only dimensionality or token-embedding scale, but also the target's readable geometry.

This view is consistent with the unreliable LS results in Table 1. Shared token embeddings can reduce a coarse space distance but still miss the compressed-vector manifold used by the target decoder. Enc-Conv helps because it can change how vectors are generated, not only map fixed source vectors.

Compressed vectors carry reusable information, but in a decoder-conditioned form. Gaussian perturbation shows that native compressed vectors occupy finite readable neighborhoods, and native geometry shows that those neighborhoods differ across decoder families. These diagnostics support the readable-region explanation without claiming a complete causal proof; direct interventions such as geometry regularization or projection remain important future tests.

7 Pre-Transfer Guidance for Retargeting

RQ3 asks whether transfer quality can be assessed before running retargeting. We evaluate Converter-only and Enc-Conv across all vector budgets and use retention (Equation 4) as the label. Predictors use only pre-transfer information: source-native evaluation, model metadata, raw target behavior, and compressed-vector geometry probes. For geometry priors, shared non-special anchors are subsampled to at most 512. The pooled score table contains more than 500 method–budget rows; Appendix E reports prior min–max summaries.

7.1 A Pair-Level Compatibility Diagnostic

The main RQ3 signal is source–target readable-geometry compatibility. Let $\mathcal{A}$ be the shared anchor set, and let $E_s(a)$ be the source decoder input embedding of anchor token a. For source vector $z_{s,i}$, define its anchor-direction profile and the target-readable anchor profile as

$$p_{s,i}(a) = \mathrm{softmax}_{a\in\mathcal{A}}\left(\frac{\cos(z_{s,i}, E_s(a))}{\tau}\right), \quad (6)$$

$$q_t(a) = \mathrm{softmax}_{a\in\mathcal{A}}\left(\frac{g_t(a)}{\tau}\right), \quad (7)$$

where $g_t(a)$ is the mean target-side anchor-direction score computed from frozen target input embeddings (Appendix D). The main diagnostic is target-readability cross-entropy:

$$\mathrm{CE}(s,t) = -\frac{1}{k}\sum_{i=1}^{k}\sum_{a\in\mathcal{A}} p_{s,i}(a)\log q_t(a). \quad (8)$$

Lower CE means source vector mass lies in anchor directions preferred by the target decoder. We orient correlations so larger values predict higher retention.

Across budgets and methods, readability CE gives oriented Spearman correlations of 27.4–47.1% with retention (min–max across settings, not confidence intervals). This is a moderate, low-cost ranking signal; Appendix E reports auxiliary RMSE and KL variants.

7.2 Guidance Depends on the Choice Being Made

The best prior depends on the decision. For a fixed target, source native NLL—the source compressor's held-out reconstruction loss under its own decoder—gives within-target $|\rho|$ of 41.3–62.9%. For a fixed source, magnitude tail counts target-standardized log-norm coordinates beyond two standard deviations and gives target-ranking $|\rho|$ of 32.7–65.5%. Formal definitions are in Appendix D.

7.3 When Is End-to-End Retargeting Needed?

The method choice depends on vector budget and target compatibility. Table 6 shows that Converter-only reaches at least 90% of Enc-Conv retention in 48.4% of matched cases, rising from 33.7% at 8 vectors to 71.4% at 32 vectors. At 32 vectors, Converter-only is sufficient in 60 of 84 directions, but Enc-Conv still gives clear gains for more restrictive targets. When the target-readable region is narrow, adapting the compressed-vector generator itself is more useful than only mapping fixed source vectors.

Setting	Reported signal	Corr. range
Across all pairs	Readability CE	27.4–47.1%
Fixed target	Source native NLL	41.3–62.9%
Fixed source	Magnitude tail	32.7–65.5%

Table 5: Pre-transfer guidance. Ranges are min–max oriented Spearman correlations with retention across methods and budgets.

k	Match Rate	Avg Gain
8	33.7%	10.4%
16	39.8%	12.5%
32	71.4%	8.1%

Table 6: Converter-only versus Enc-Conv on matched transfer directions. **Match Rate** denotes the fraction of directions where Converter-only achieves at least $0.9\times$ Enc-Conv retention. **Avg Gain** is the mean absolute retention gain of Enc-Conv over Converter-only (percentage points on the retention scale).

Overall, RQ3 shows that decoder-interface portability is partly assessable before running a full retargeting experiment. Readability CE gives the main all-pair diagnostic signal, source-native quality helps choose among compressors for a fixed target, and tail geometry helps choose targets for a fixed source.

8 Conclusion

We study soft-compressor retargeting as decoder-interface portability. Compressed vectors contain reusable information, but their usability depends on the target decoder's continuous input space. Random vectors and LS alignment are unreliable; trainable retargeting, especially Enc-Conv, can recover much of the target-native reconstruction quality in favorable settings.

Transfer is uneven because retargeting must preserve source content and place vectors in the target-readable region. This explains target dependence, directional asymmetry, and the value of adapting the generator. Matched-budget scratch shows Enc-Conv is not merely a budget-matched native retrain, though some targets still let scratch compete; pair-level readable-geometry scores give only moderate pre-transfer guidance.

Limitations

This work focuses on controlled retargeting methods rather than exhausting transfer algorithms. Current linear and anchor-style alignments are insufficient, but stronger maps, richer converters, cross-token modules, or lightly unfreezing early target layers may narrow the gaps.

We use reconstruction on 128-token segments as a controlled proxy for decoder-interface readability, not as evidence for downstream QA, RAG, summarization, or reasoning. Runs use one seed and no confidence intervals, so small BLEU gaps and moderate correlations are descriptive. Perturbation and geometry analyses support the target-readable-region view, while intervention tests such as geometry regularization remain future work.

References

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3829–3846, Singapore. Association for Computational Linguistics.

Yuhong Dai, Jianxun Lian, Yitian Huang, Wei Zhang, Mingyang Zhou, Mingqi Wu, Xing Xie, and Hao Liao. 2025. Pretraining context compressor for large language models with embedding-based memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28715–28732, Vienna, Austria. Association for Computational Linguistics.

Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Rishi Jha, Collin Zhang, Vitaly Shmatikov, and John Morris. 2025. Harnessing the universal geometry of embeddings. In *Advances in Neural Information Processing Systems*, volume 38, pages 45963–45987. Curran Associates, Inc.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Sam Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Guillaume Lengyel, Guillaume Lample, Lucaille Saulnier, Pierre-Emmanuel Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Pavicic, Thomas Malartre, Vladimir Stojnic, Edouard Douxchamps, and Guillaume Roux. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

Yuri Kuratov, Mikhail Arkhipov, Aydar Bulatov, and Mikhail Burtsev. 2025. Cramming 1568 tokens into a single vector and back again: Exploring the limits of embedding space capacity. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19323–19339, Vienna, Austria. 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.

Gleb Mezentsev and Ivan Oseledets. 2025. Exploring the hidden capacity of LLMs for one-step text generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22880–22889, Suzhou, China. Association for Computational Linguistics.

Antonio Norelli, Marco Fumero, Valentino Maiorca, Luca Moschella, Emanuele Rodola, and Francesco Locatello. 2024. Asif: Coupled data turns unimodal models to multimodal without training. *Advances in Neural Information Processing Systems*, 36.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981, Bangkok, Thailand. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, pages 30811–30849. Curran Associates, Inc.

Hippolyte Pilchen, Edouard Grave, and Patrick Pérez. 2025. Arc-encoder: learning compressed text representations for large language models. *Preprint*, arXiv:2510.20535.

Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, and Jie Zhou. 2022. On transferability of prompt tuning for natural language processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3949–3969, Seattle, United States. Association for Computational Linguistics.

Xiangfeng Wang, Zaiyi Chen, Tong Xu, Zheyong Xie, Yongyi He, and Enhong Chen. 2024. In-context former: Lightning-fast compressing context for large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2445–2460, Miami, Florida, USA. Association for Computational Linguistics.

Zhaofeng Wu, Xinyan Yu, Dani Yogatama, Jiasen Lu, and Yoon Kim. 2025. The semantic hub hypothesis: Language models share semantic representations across languages and modalities. In *International Conference on Learning Representations*, volume 2025, pages 53705–53723.

Zijun Wu, Yongkang Wu, and Lili Mou. 2024. Zero-shot continuous prompt transfer: Generalizing task semantics across language models. In *International Conference on Learning Representations*, volume 2024, pages 1482–1496.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Jinsung Yoon and Sercan O Arik. 2025. Embedding-converter: A unified framework for cross-model embedding transformation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25464–25482, Vienna, Austria. Association for Computational Linguistics.

A Hyperparameters

Tables 7 and 8 summarize the decoder set and shared training settings; Table 9 reports median wall time and GPU-hours for the timed training configurations.

Decoder	**Abbr.**	**Params**	d
Llama-3.2-1B-Instruct	LLaMA-1B	1B	2048
Llama-3.2-3B-Instruct	LLaMA-3B	3B	3072
Llama-3-8B-Instruct	LLaMA-8B	8B	4096
Qwen2.5-1.5B-Instruct	Qwen-1.5B	1.5B	1536
Qwen2.5-3B-Instruct	Qwen-3B	3B	2048
Qwen2.5-7B-Instruct	Qwen-7B	7B	3584
Mistral-7B-Instruct	Mistral-7B	7B	4096

Table 7: Target decoders and their hidden size d. The Abbr. column lists the shorthand names used throughout the paper. Transfer uses frozen encoder backbones GPT-2 (d=768) and LLaMA-1B (d=2048).

Setting	**Value**
Data / Architecture	
Segment length	128 tokens
Compressed vectors k	$\{8, 16, 32\}$
Corpus	FineWeb-10BT
Transfer train data	$0.1\times$ Origin train set
Backbone	`GPT-2` or `LLaMA-1B`
Target decoder	Frozen
Converter MLP	RMSNorm + Linear $d_c \to d_t$; GELU + Linear $d_t \to d_t$
Trainable modules	Converter-only or Enc-Conv
Optimization / Runtime	
Optimizer	AdamW
Learning rate	10^{-4} with cosine schedule
Weight decay	0.01
Warmup ratio	0.1 (transfer)
Epochs / precision	1 epoch / bf16
Batch sizes	Origin: 32/8; transfer: 16 per GPU
Hardware	ZeRO-1 on $4\times$A800 80G
Random seed	42
Evaluation / Baselines	
Held-out examples	1,000
Decoding	Greedy decoding; max 128 tokens
Acc / BLEU	Next-token accuracy / autoregressive BLEU
Random / LS	Gaussian $\mathcal{N}(0, 0.02^2)$ / shared-vocabulary map

In all transfer runs, the target decoder is frozen and reconstruction loss is computed on target-token likelihood.

Table 8: Training and evaluation hyperparameters.

Setting	n	**Median wall**	**Median GPU-h**
ORIGIN, `GPT-2`, $k\in\{8, 16, 32\}$	21	2.1 h	8.4
ORIGIN, `LLaMA-1B`, $k\in\{8, 16, 32\}$	21	4.8 h	19.2
Conv-only, k=8/16/32	98 each	12.9/13.7/16.4 min	0.86/0.91/1.09
Enc-Conv, k=8/16/32	98 each	27.6/28.3/30.9 min	1.84/1.89/2.06
Total reported configs	630	≈383 h	≈1,534

Table 9: Median training time on four A800 GPUs (ZeRO-1). **GPU-h** sums GPU–hours across the node as $4\times$ wall hours because each job uses four GPUs in parallel (HuggingFace `train_runtime`). Random/LS baselines and evaluation sweeps omit timed optimization.

Pair	LS	Procrustes	Anchor-BP
LLaMA-1B→LLaMA-8B	0.033	0.000	0.024
LLaMA-1B→Mistral-7B	0.027	0.002	0.027
LLaMA-1B→Qwen-1.5B	0.095	0.089	0.118
LLaMA-1B→Qwen-3B	0.137	0.130	0.142
LLaMA-3B→Qwen-3B	0.132	0.144	0.142
Qwen-1.5B→LLaMA-3B	0.000	0.000	0.012

Table 10: Anchor and linear baselines at 512 anchors (BLEU). Anchor-BP follows the anchor-search idea of zero-shot continuous prompt transfer (Wu et al., 2024), using 50 evaluation samples and 500 search iterations per vector. Results remain far below trainable retargeting in Table 1.

Pair	full	32	128	512	1024	2048
LLaMA-1B→LLaMA-3B	0.030	0.000	0.031	0.042	0.010	0.000
LLaMA-1B→LLaMA-8B	0.038	0.000	0.030	0.033	0.000	0.000
LLaMA-1B→Mistral-7B	0.009	0.000	0.000	0.027	0.016	0.000
LLaMA-1B→Qwen-1.5B	0.093	0.107	0.110	0.095	0.047	0.007
LLaMA-1B→Qwen-3B	0.137	0.130	0.136	0.137	0.124	0.004
LLaMA-3B→Qwen-3B	0.147	0.132	0.129	0.132	0.136	0.150
Qwen-1.5B→LLaMA-3B	0.000	0.001	0.000	0.000	0.000	0.000
Qwen-3B→LLaMA-8B	0.000	0.016	0.000	0.000	0.000	0.000

Table 11: LS BLEU under different shared-anchor caps. "full" uses all single-token anchors from the main LS setup; numbered columns cap the anchor set before fitting the linear map. Increasing anchor count does not consistently improve reconstruction.

B Implementation Details

Converter-only trains only the RMSNorm–MLP converter; Enc-Conv also updates the compressor encoder, with the target decoder frozen. LS fits a shared-anchor embedding map:

$$W_{\mathrm{LS}} = \arg\min_{W} \|E_s(\mathcal{A})W - E_t(\mathcal{A})\|_F^2 . \quad (9)$$

$$Z_t = Z_s W_{\mathrm{LS}}. \quad (10)$$

Here E_s, E_t are source and target input-embedding lookup tables.

RQ3 anchors are

$$\mathcal{A} = \mathrm{subsample}_{\leq 512}\left(V_s \cap V_t \setminus \mathcal{S}\right), \quad (11)$$

where V_s, V_t are vocabularies and $\mathcal{S}$ special tokens. We use $\tau = 0.1$ for anchor-direction softmaxes; Section 7 defines CE.

C Anchor and Linear Baselines

Table 10 reports anchor checks. Anchor-BP optimizes anchor-relative coordinates (Wu et al., 2024); Procrustes uses an orthogonal centered map. Both track LS rather than trainable retargeting.

Table 11 varies LS anchor count. More anchors are not reliably better, suggesting that token-embedding alignment is a poor proxy for the compressed-vector interface.

D Metric and Geometry Definitions

BLEU. Let y be the generated string and r the reference for the 128-token segment. We use NLTK `wordpunct_tokenize` and sentence BLEU with Chen–Cherry smoothing method 1. If lengths differ, the shorter sequence is matched to the best equal-length span of the longer one. **Acc** follows Section 4.4.

Target anchor-direction scores and pair metrics. Let $p_{s,i}$ and q_t be the anchor softmaxes from Section 7. For each anchor $a \in \mathcal{A}$, $g_t(a)$ is the mean target-side anchor-direction score computed from frozen target input embeddings (used in q_t). With $\delta_{s,i}(a) = \cos(z_{s,i}, E_s(a)) - g_t(a)$, pair direction RMSE is

$$\mathrm{RMSE}_{\mathrm{dir}} = \left(\frac{1}{k|\mathcal{A}|}\sum_{i,a}\delta_{s,i}(a)^2\right)^{\frac{1}{2}}. \quad (12)$$

Anchor KL is $\frac{1}{k}\sum_i \mathrm{KL}(p_{s,i} \,\|\, q_t)$.

Source native NLL. Source native NLL is held-out reconstruction loss under the source decoder, with $Z_s = \mathrm{Compressor}_s(x)$:

$$\mathrm{NLL}_s = -\frac{1}{|\mathcal{D}|}\sum_{(x,\mathbf{y})\in\mathcal{D}} \frac{\log p_{D_s}(\mathbf{y} \mid Z_s)}{|\mathbf{y}|}. \quad (13)$$

Here $\mathbf{y}$ is the gold target-token sequence for segment x.

Tail fractions. For direction scores or log-norm magnitudes, standardize each coordinate u_j by target-anchor statistics:

$$z_j = \frac{u_j - \mu_{t,j}}{\max(\sigma_{t,j}, 10^{-3})}. \quad (14)$$

$$\mathrm{Tail} = \frac{1}{m}\sum_{j=1}^{m} \mathbf{1}\{|z_j| > 2\}. \quad (15)$$

We average this over compressed-vector slots and probe texts, where m is the number of standardized coordinates in the probe.

E Additional Results

This section collects supporting tables: GPT-2-source transfer (Table 12), compressed-vector length (Table 13), and RQ3 prior min–max summaries (Table 14).

Block	Transfer Pair	Random	LS	Converter	Enc-Conv	Origin
A → A	LLaMA-1B → Qwen-1.5B	0.404/0.127$_{-62\%}$	0.360/0.097$_{-71\%}$	0.610/0.097$_{-71\%}$	**0.654/0.173**$_{-48\%}$	0.916/0.334
	Qwen-1.5B → LLaMA-1B	0.354/0.092$_{-89\%}$	0.004/0.000$_{-100\%}$	0.888/0.662$_{-23\%}$	**0.897/0.695**$_{-19\%}$	0.955/0.860
A → B	LLaMA-1B → Qwen-3B	0.323/0.093$_{-88\%}$	0.418/0.132$_{-84\%}$	0.462/0.031$_{-96\%}$	**0.719/0.272**$_{-66\%}$	0.945/0.801
	LLaMA-1B → LLaMA-3B	0.369/0.100$_{-88\%}$	0.256/0.034$_{-96\%}$	0.943/0.835$_{-3\%}$	**0.945/0.850**$_{-1\%}$	0.960/0.862
	Qwen-1.5B → Qwen-3B	0.323/0.093$_{-88\%}$	0.420/0.146$_{-82\%}$	0.480/0.038$_{-95\%}$	**0.738/0.293**$_{-63\%}$	0.945/0.801
	Qwen-1.5B → LLaMA-3B	0.369/0.100$_{-88\%}$	0.008/0.000$_{-100\%}$	0.917/0.761$_{-12\%}$	**0.924/0.784**$_{-9\%}$	0.960/0.862
A → C	LLaMA-1B → LLaMA-8B	0.410/0.116$_{-87\%}$	0.152/0.021$_{-98\%}$	0.963/0.877$_{-1\%}$	**0.965/0.880**$_{-1\%}$	0.969/0.889
	LLaMA-1B → Mistral-7B	0.423/0.103$_{-89\%}$	0.193/0.012$_{-99\%}$	0.975/0.886$_{-7\%}$	**0.978/0.900**$_{-6\%}$	0.990/0.954
	LLaMA-1B → Qwen-7B	0.431/0.128$_{-83\%}$	0.443/0.134$_{-82\%}$	0.918/0.745$_{+0\%}$	**0.932/0.765**$_{+3\%}$	0.954/0.742
	Qwen-1.5B → LLaMA-8B	0.409/0.115$_{-87\%}$	0.003/0.000$_{-100\%}$	0.942/0.808$_{-9\%}$	**0.950/0.831**$_{-7\%}$	0.969/0.889
	Qwen-1.5B → Mistral-7B	0.423/0.103$_{-89\%}$	0.080/0.000$_{-100\%}$	0.959/0.853$_{-11\%}$	**0.967/0.874**$_{-8\%}$	0.990/0.954
	Qwen-1.5B → Qwen-7B	0.432/0.129$_{-83\%}$	0.430/0.124$_{-83\%}$	0.914/0.718$_{-3\%}$	**0.918/0.740**$_{-0\%}$	0.954/0.742
B → B	Qwen-3B → LLaMA-3B	0.369/0.100$_{-88\%}$	0.059/0.000$_{-100\%}$	**0.930/0.813**$_{-6\%}$	0.936/0.804$_{-7\%}$	0.960/0.862
	LLaMA-3B → Qwen-3B	0.323/0.093$_{-88\%}$	0.421/0.144$_{-82\%}$	0.721/0.232$_{-71\%}$	**0.782/0.361**$_{-55\%}$	0.945/0.801
B → C	Qwen-3B → LLaMA-8B	0.409/0.128$_{-86\%}$	0.000/0.000$_{-100\%}$	0.953/0.824$_{-7\%}$	**0.957/0.846**$_{-5\%}$	0.969/0.889
	Qwen-3B → Mistral-7B	0.423/0.103$_{-89\%}$	0.032/0.000$_{-100\%}$	0.967/0.843$_{-12\%}$	**0.972/0.887**$_{-7\%}$	0.990/0.954
	Qwen-3B → Qwen-7B	0.432/0.164$_{-78\%}$	0.433/0.150$_{-80\%}$	**0.929/0.786**$_{+6\%}$	0.930/0.765$_{+3\%}$	0.954/0.742
	LLaMA-3B → LLaMA-8B	0.409/0.128$_{-86\%}$	0.155/0.035$_{-96\%}$	0.957/0.848$_{-5\%}$	**0.956/0.852**$_{-4\%}$	0.969/0.889
	LLaMA-3B → Mistral-7B	0.423/0.103$_{-89\%}$	0.135/0.006$_{-99\%}$	0.966/0.844$_{-12\%}$	**0.969/0.869**$_{-9\%}$	0.990/0.954
	LLaMA-3B → Qwen-7B	0.432/0.164$_{-78\%}$	0.438/0.167$_{-77\%}$	0.908/0.687$_{-7\%}$	**0.914/0.711**$_{-4\%}$	0.954/0.742
C → C	LLaMA-8B → Mistral-7B	0.423/0.103$_{-89\%}$	0.095/0.000$_{-100\%}$	**0.956/0.838**$_{-12\%}$	0.955/0.816$_{-14\%}$	0.990/0.954
	LLaMA-8B → Qwen-7B	0.432/0.164$_{-78\%}$	0.436/0.163$_{-78\%}$	0.900/0.668$_{-10\%}$	**0.910/0.697**$_{-6\%}$	0.954/0.742
	Mistral-7B → LLaMA-8B	0.409/0.128$_{-86\%}$	0.013/0.000$_{-100\%}$	0.928/0.799$_{-10\%}$	**0.947/0.814**$_{-9\%}$	0.969/0.889
	Mistral-7B → Qwen-7B	0.432/0.164$_{-78\%}$	0.319/0.083$_{-89\%}$	0.894/0.696$_{-6\%}$	**0.925/0.756**$_{+2\%}$	0.954/0.742
	Qwen-7B → LLaMA-8B	0.409/0.128$_{-86\%}$	0.031/0.000$_{-100\%}$	0.952/0.827$_{-7\%}$	**0.955/0.846**$_{-5\%}$	0.969/0.889
	Qwen-7B → Mistral-7B	0.423/0.103$_{-89\%}$	0.139/0.004$_{-100\%}$	0.967/0.858$_{-10\%}$	**0.970/0.872**$_{-9\%}$	0.990/0.954

Table 12: GPT-2 compressor transfer results at the 32-vector setting. Same notation as Table 1.

Setting	8 vec	16 vec	32 vec
Origin Acc. (LLaMA-1B source)	0.925	0.981	0.997
Origin BLEU (LLaMA-1B source)	0.577	0.788	0.790
Origin Acc. (GPT-2 source)	0.762	0.887	0.956
Origin BLEU (GPT-2 source)	0.319	0.552	0.622
Converter BLEU	0.236	0.406	0.517
Enc-Conv BLEU	0.272	0.482	0.603

Table 13: Effect of compressed-vector length. Origin rows average over target-native compressors; Converter and Enc-Conv rows average over transfer evaluations.

Prior	Global ρ	Group signal
Readability CE	27.4–47.1%	T2 33.9–46.2%
Pair direction RMSE	23.1–49.9%	T2 34.3–50.0%
Source→target KL	23.7–46.3%	T2 34.3–49.5%
Magnitude tail	17.2–26.7%	T2 32.7–65.5%
Source NLL	3.1–26.8%	T1 41.3–62.9%
Same family	1.2–19.3%	T1 26.5–52.5%

Table 14: RQ3 prior *ranges* report the minimum and maximum oriented Spearman $|\rho|$ with retention when pooling Converter-only and Enc-Conv and all vector budgets (not confidence intervals). T1 ranks sources for a fixed target; T2 ranks targets for a fixed source compressor.