WORD-LEVEL TEXT UNMIXING VIA EVIDENCE-PRESERVING OWNERSHIP ROUTING WITH LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Text from multiple sources can become interleaved into a single sequence when attribution metadata is lost, such as overlapping speech transcripts, complex document reading flows, or concurrent agent streams. While all observed words remain present, their underlying source ownership is unknown. We formalize this challenge as **Word-Level Text Unmixing**: given an interleaved lexical stream and the source count K, recover the original source sequences while preserving every word occurrence and its within-source order exactly. Conventional large language model (LLM) paradigms approach this by generating separated source texts directly. However, coupling latent ownership inference with lexical regeneration frequently causes LLMs to omit, duplicate, or hallucinate words, violating the exact-reconstruction objective. To resolve this, we propose **Evidence-Preserving Ownership Routing (EPOR)**, a framework that clearly separates ownership prediction from sequence reconstruction. During training, EPOR adapts a causal LLM to predict canonical ownership routes conditioned on the interleaved stream and prior routing decisions, inducing dynamic, history-conditioned source contexts without lexical regeneration. At inference, EPOR pairs completion-safe constrained decoding with deterministic indexed reconstruction, guaranteeing structurally valid K-source partitions that preserve every observed token occurrence exactly once. For systematic evaluation, we introduce **UNMIXBENCH**, spanning controlled synthetic mixtures, timestamp-derived speech from AMI and ICSI, layout-derived document streams from ReadingBank, and simulated concurrent digital outputs. Across five evaluation tracks, a 4B EPOR model achieves the lowest minimum-permutation word error rate (MP-WER) among finetuned baselines, reducing the five-track mean by 22.3% relative to compact source-array generation and competing favorably with zero-shot frontier LLMs. These findings indicate that when lexical evidence is fully observed, decoupling ownership routing from sequence reconstruction can provide a more reliable and structurally sound alternative to generative decoders.

1 INTRODUCTION

Text from multiple sources frequently becomes interleaved into a single sequence when attribution metadata is lost during downstream processing. Although the observed words remain intact, the mapping between each word occurrence and its original source is unknown. Recovering these latent source assignments is critical for restoring readable transcripts, coherent document structures, and interpretable digital records across several primary settings. In **temporal interleaving** for speech processing, transcribed words from multiple speakers merge into a single chronological stream due to overlaps or interruptions, making source attribution essential for speaker-aware analysis (Paturi et al., 2023; Wang et al., 2024). In **spatial interleaving**, document extraction systems often serialize two-dimensional layouts like multi-column pages into an incorrect linear order (Wang et al., 2021; Zhang et al., 2023). Furthermore, **concurrent interleaving** occurs in digital systems when collaborative agents or unlabelled event logs emit partial outputs into a shared stream where only the arrival order is preserved (Wu et al., 2024; Wang et al., 2025).

All these scenarios share a fundamental structure where the observed words are available but their source ownership is unknown. We formalize this problem as **Word-Level Text Unmixing**, as shown in

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

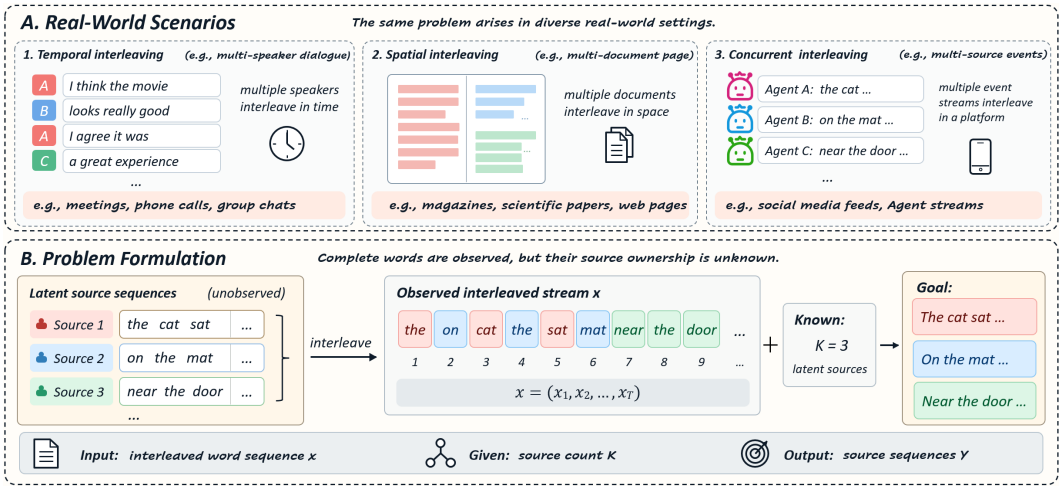


Figure 1: The **Word-Level Text Unmixing** problem. Temporal, spatial, and concurrent serialization motivate recovery from an interleaved lexical stream. Given the observed words and known source count K, recover the latent sources while preserving every occurrence and its within-source order.

Figure 1. Given a completely interleaved word stream and source count K, the objective is to recover the distinct source sequences such that every observed word occurrence is used exactly once and its original within-source order is preserved. Unlike coarser text separation tasks like utterance-level conversation disentanglement (Elsner & Charniak, 2008; Kummerfeld et al., 2019), paragraph-level style-change detection (Zangerle et al., 2022), or temporally grounded speaker diarization (Paturi et al., 2023; Wang et al., 2024), word-level text unmixing requires resolving ownership shifts between adjacent words using solely the mixed lexical stream.

While generative large language models (LLMs) are commonly used for text-to-structure extraction (Lu et al., 2022; Wang et al., 2023; Sainz et al., 2024), directly prompting them to generate separated source arrays presents two fundamental limitations for word-level text unmixing. First, because each ownership assignment alters the partial source contexts needed to interpret subsequent words, models must perform history-conditioned routing decoupled from lexical reconstruction. Second, standard free-generation paradigms couple ownership inference with text generation. This allows structurally correct ownership deductions to be corrupted by the omission, duplication, or alteration of observed words, directly violating the exact-reconstruction objective.

To resolve this, we introduce **Evidence-Preserving Ownership Routing (EPoR)**, a framework that decouples learned source ownership from deterministic reconstruction. During training, a causal language model predicts a single ownership label per observed word, conditioned on the interleaved stream and prior routing decisions. This enables history-conditioned ownership prediction rather than prone-to-error lexical regeneration. At inference, EPoR a completion-safe constrained beam search to evaluate candidate routes, ensuring the selected paths can form complete K-source partitions. Indexed reconstruction then mechanically gathers the original word occurrences based on these routes, enforcing structural validity and exact evidence preservation.

For systematic evaluation, we construct **UnMixBench**, a benchmark covering controlled synthetic mixtures, timestamp-derived speech from AMI and ICSI, layout-derived reading flows from ReadingBank, and simulated concurrent digital streams via WikiText. Across the five principal evaluation tracks, a 4B EPoR model achieves the lowest mean minimum-permutation word error rate (MP-WER) among task-finetuned free-generation and structured baselines, yielding a 22.3% error reduction relative to compact source-array generation. Furthermore, EPoR remains highly competitive with zero-shot frontier language models.

Our key contributions include the following.

- We formalize **word-level text unmixing**, a fine-grained de-interleaving problem aimed at recovering the ownership of each word occurrence from mixed text.

- We propose Evidence-Preserving Ownership Routing (EPoR), a structured framework that decouples history-conditioned source ownership prediction from evidence-preserving reconstruction.
- We introduce **UnMixBench** to evaluate text unmixing across temporal, spatial, and concurrent serialization scenarios, demonstrating EPoR's superior performance across multiple domains.

2 Related Work

Message- and utterance-level text disentanglement. In conversation disentanglement, most methods use lexical context, speaker and temporal metadata, mentions, reply structure, and learned representations (Elsner & Charniak, 2010; Mehri & Carenini, 2017; Jiang et al., 2018; Kummerfeld et al., 2019; Yu & Joty, 2020). Extensions consider zero-shot or weakly supervised objectives and explicit speaker or reference structure (Chi & Rudnicky, 2021; Zhu et al., 2021; Huang et al., 2022; Ma et al., 2022). These settings retain intact linguistic units and often expose source-related metadata (Chang et al., 2023; Gouravajhala et al., 2023). Style-change detection similarly locates authorship changes at paragraph or sentence boundaries using stylistic evidence (Zangerle et al., 2022). Text unmixing instead reconstructs sources from individually interleaved word occurrences.

Lexical and multimodal speaker binding. Speaker separation and diarization use waveforms, timing, spatial cues, and learned speaker representations; multimodal systems also use lexical context (Chen et al., 2017; Horiguchi et al., 2020; Park & Georgiou, 2018; Yin et al., 2025; Lin et al., 2025). Lexical variants and language-model post-processing show that words can inform fine-grained speaker assignment or correct upstream diarization labels (Park & Georgiou, 2018; Paturi et al., 2023; Park et al., 2024; Wang et al., 2024). These systems typically receive acoustic evidence or the initial speaker hypothesis. Instead, our task recovers the latent sources without acoustic cues or the upstream speaker hypothesis and requires exact preservation of the observed word occurrences. Related formulations also arise beyond NLP, including shuffle recognition and string unshuffling, document reading-order recovery, and blind source separation. We discuss these broader connections and their distinctions from word-level text unmixing in Appendix E.

Exchangeable structured prediction and constrained generation. Permutation-invariant training optimizes over equivalent label assignments (Yu et al., 2017); autoregressive clustering conditions assignments on earlier labels (Pakman et al., 2020); restricted-growth encodings select one representative per unlabeled partition (Milne, 1982); and constrained generation restricts outputs to a prescribed structure (Hokamp & Liu, 2017; Post & Vilar, 2018; Scholak et al., 2021; Geng et al., 2023). These methods address label symmetry, assignment dependence, and output constraints as general prediction problems.

3 Problem Formulation

Definition. In word-level text unmixing, the input consists of a sequence of indexed word occurrences $x=(x_1,...,x_T)$ and a known source count K, with $1 \le K \le T$. A word occurrence is an indivisible, corpus-defined task unit whose encoding may contain multiple LLM tokens. Let $z^\star$ denote the latent generating ownership route. For any candidate route $z=(z_1,...,z_T)$, the label $z_t=k$ assigns occurrence x_t to source k. Defining $[K]=\{1,...,K\}$, the labeled assignment space requires surjectivity so that every source is non-empty:

$$\mathcal{P}^{\mathrm{lab}}_{T,K} = \big\{ z \in [K]^T : \{z_1,...,z_T\} = [K] \big\}. \tag{1}$$

Stable filtering reconstructs source k by reading its assigned occurrences in increasing input order:

$$R_k(x,z) = (x_t : z_t = k,\, t = 1,...,T) \tag{2}$$

Writing $R(x,z) = (R_1(x,z),...,R_K(x,z))$, the generating source tuple is $Y^\star = R(x,z^\star)$. Source labels are exchangeable, meaning that for any permutation in the symmetric group S_K, the tuples $Y = (y_1,...,y_K)$ and $Y^\pi = (y_{\pi(1)},...,y_{\pi(K)})$ represent the same answer. The text unmixing task therefore recovers $(x,K) \longmapsto [Y^\star]$, where $Y^\star = (y^\star_1,...,y^\star_K)$, and brackets denote equivalence under a global source permutation. The output is an unordered collection with multiplicity rather than a set of named sources, and its valid output space is:

$$\mathcal{Y}_{\mathrm{valid}}(x,K) = \big\{ [R(x,z)] : z \in \mathcal{P}^{\mathrm{lab}}_{T,K} \big\}. \tag{3}$$

The only input is (x, K), and no source identities, timestamps, layout coordinates, or upstream attribution hypotheses are available, while the word occurrences themselves are complete and fixed.

Example. Consider two latent source sequences, "the volcano erupts" and "the chef cooks." After interleaving, the model observes only the indexed sequence

$$x = (\text{the}, \text{the}, \text{volcano}, \text{erupts}, \text{chef}, \text{cooks})$$

alongside the number of sources $K = 2$. The word-level text unmixing task is to recover the two original source sequences by determining which source owns each observed word occurrence, while using every occurrence exactly once and preserving its order within that source. For this scenario, the latent ownership route is

$$z = (1,2,1,1,2,2),$$

which groups (x_1, x_3, x_4) into "the volcano erupts" and (x_2, x_5, x_6) into "the chef cooks." An ownership route is a sequence of source labels for the observed occurrences. The model does not observe z but rather predicts this auxiliary representation in order to reconstruct the sources.

Validity and correctness. Structural validity strictly requires exactly K non-empty sources, exact evidence preservation, and unchanged word order within each assigned source. Evidence preservation dictates that every observed word occurrence is reused exactly once. Semantic correctness evaluates the logical and semantic validity of each resulting unmixed source sequence.

4 Approach: Evidence-Preserving Ownership Routing

EPoR separates learned source ownership from evidence preservation. During training, a causal LLM predicts an ownership label for each observed word, conditioned on the complete mixed stream, the source count, and preceding labels. These labels define partial source contexts that guide subsequent predictions. At inference, the model scores candidate ownership routes, while completion-safe beam search retains prefixes that can form a partition into K non-empty sources. Indexed reconstruction then selects the original occurrences in source-local order. Figure 2 shows both stages.

4.1 Training: Ownership-Only Adaptation

For each example $(x, K, z^\star)$, we canonicalize the gold route by source order of first appearance and adapt the causal LLM with QLoRA under teacher-forced negative log-likelihood. $z^\star$ denotes the canonical gold route. Algorithm 1 in Appendix C.7 gives the procedure.

4.1.1 Canonical Representation

Because source labels are exchangeable, we use a first-appearance canonical ownership representation. The source containing the earliest occurrence is labeled 1, the next unseen source is labeled 2, and so on. This restricted-growth representation (Milne, 1982) defines

$$\mathcal{Z}_{T,K} = \left\{ \begin{array}{l} z \in \{1,\ldots,K\}^T : \; z_1 = 1, \\ z_t \le 1 + \max_{j<t} z_j \text{ for } t > 1, \quad \max_t z_t = K \end{array} \right\}. \tag{4}$$

Each unlabeled partition into K non-empty sources has one such route, replacing $K!$ equivalent labelings with a shared target for supervision and decoding. This removes naming redundancy without resolving semantic ambiguity (Appendix C.4).

4.1.2 History-Conditioned Ownership Adaptation

The learning target is the canonical ownership route $z^\star$. Given the complete indexed stream x and source count K, the LLM predicts one ownership label per word occurrence using its existing vocabulary head.

The training scorer factorizes autoregressively:

$$s_\theta(z; x, K) = \sum_{t=1}^{T} \log p_\theta(z_t \mid x, K, z_{<t}). \tag{5}$$

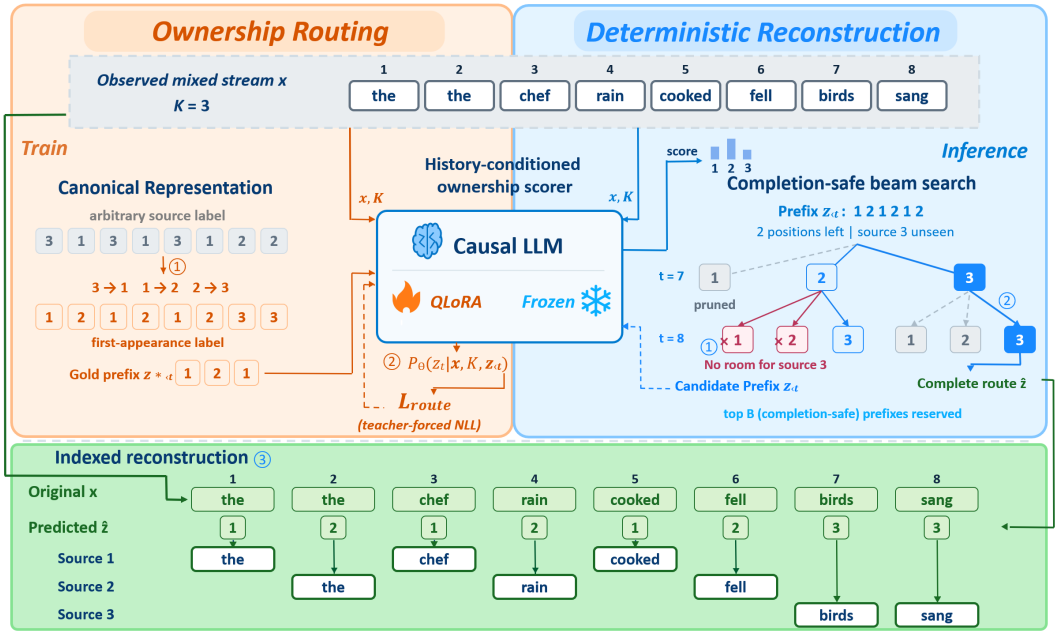


Figure 2: **Evidence-Preserving Ownership Routing (EPOR). Training:** the causal LLM learns canonical ownership from the complete mixed input and gold prefixes. **Inference:** completion-safe beam search scores candidate ownership routes, and then indexed reconstruction gathers the original occurrences into sources.

Here $p_\theta(z_t \mid \cdot)$ is the full-vocabulary LLM probability of the symbol encoding z_t. During training, each label prediction is conditioned on the gold route prefix $z^\star_{<t}$. Together with x, a prefix determines the partial sequence for each source,

$$H_{k,t}(x, z_{<t}) = (x_j : j < t,\, z_j = k). \tag{6}$$

Different prefixes associate the same occurrence with different source contexts. The sequences $H_{k,t}$ are implicit in $(x, z_{<t})$, rather than separate model inputs. As in autoregressive clustering (Pakman et al., 2020), the scorer can use the contexts implied by earlier assignments.

Let P_i be the prompt containing (x_i, K_i), $v_i^\star$ the serialized canonical route with its assistant termination suffix, and $\mathcal{U}_i$ the supervised LM-token positions. The QLoRA adaptation objective is

$$\mathcal{L}_{\text{route}}(\theta) = -\frac{1}{\sum_i |\mathcal{U}_i|} \sum_i \sum_{u \in \mathcal{U}_i} \log p_\theta(v^\star_{i,u} \mid P_i, v^\star_{i,<u}). \tag{7}$$

The prompt is loss-masked, and the route and assistant termination suffix are supervised (Appendix C.4). Because the complete x precedes the route, every decision can attend to the full mixed stream.

4.2 Inference: Constrained Ownership Routing

At inference, the adapted causal LLM scores each candidate ownership route using its predicted prefix. Completion-safe constrained decoding searches routes that can form K non-empty sources, and then indexed reconstruction then selects the original word occurrences according to the chosen route. Algorithm 2 in Appendix C.7 gives the complete procedure.

4.2.1 Completion-Safe Decoding

First-appearance validity alone does not ensure that all K sources can appear before the route ends. Before position t, let $m_{t-1} = \max z_{<t}$, with $m_0 = 0$. The admissible labels are

$$\mathcal{A}_t(z_{<t}) = \{a \in \{1, \ldots, \min(K, m_{t-1}+1)\} : K - \max(m_{t-1}, a) \le T - t\}. \tag{8}$$

Table 1: **The UNMIXBENCH evaluation tracks.** Counts give examples / provenance groups. K is the source count and T the number of observed word occurrences.

Track	Corpus	Examples / groups	K	T	Interleaving
Controlled	MultiNLI	2,640 / 240	2–4	16/24/32	Controlled intensity
Temporal	AMI	1,000 / 59	2/3	64	Word timestamps
Temporal	ICSI dev	400 / 15	2/3	64	Word timestamps
Spatial	ReadingBank	833 / 828	2–4	32/64	Reading-flow metadata
Concurrent	WikiText	1,200 / 400	2–4	32/64	Virtual event times

The range admits an existing source or the next unseen source, so that the remaining-capacity condition reserves enough positions for all missing sources. If two positions and two unseen sources remain, the next label must introduce a source. Appendix C.5 proves that these conditions characterize exactly the extensions admitting a complete canonical route.

4.2.2 FINITE-BEAM OWNERSHIP SEARCH

Each prefix is expanded over $\mathcal{A}_t(z_{<t})$, retaining the top B extensions under Equation 5. As in constrained decoding (Hokamp & Liu, 2017; Post & Vilar, 2018; Scholak et al., 2021; Geng et al., 2023), illegal actions are masked. Scores retain full-vocabulary log probabilities without renormalization over legal labels. Decoding stops after T decisions.

The search retains the top B extensions at each step:

$$\widehat{z}_B = \mathrm{BeamSearch}_B(s_\theta; \mathcal{Z}_{T,K}). \tag{9}$$

Here B is the beam width.

4.2.3 OCCURRENCE-INDEXED RECONSTRUCTION

Stable filtering $R(x,\widehat{z}_B)$ in Equation 2 reads each original occurrence exactly once, in an increasing input order within its assigned source. Together with completion-safe decoding, it yields K non-empty source sequences after T route decisions:

$$\widehat{Y} = R(x,\widehat{z}_B), \tag{10}$$

where R is the stable-filtering map. Indexed reconstruction preserves the observed occurrences regardless of whether the predicted ownership route is correct. Appendix C.6 gives the formal guarantee.

5 UNMIXBENCH

UNMIXBENCH is a benchmark introduced to evaluate word-level text unmixing across five text-interleaving tracks. Each example contains a mixed stream x, source count K, reference sources $Y^\star$, and gold ownership route $z^\star$. Models receive only x and K, while source identities and construction metadata are withheld. Task adaptation and checkpoint selection use 10,000 training and 1,000 development mixtures from disjoint SNLI sources.

Five evaluation tracks. Table 1 summarizes the tracks used in the main comparison. Controlled MultiNLI varies source count K, total length T, mixing intensity M, and source similarity S on held-out source groups. Here, *controlled* describes the mixture construction, and evaluation still involves a corpus shift from SNLI. Temporal tracks merge AMI and ICSI words by annotated times. The spatial track derives reading-flow segments from ReadingBank XML metadata (Wang et al., 2021). Concurrent digital serialization applies smooth, bursty, and unequal-rate virtual event schedules to the same 400 source groups and averages the three profiles. Appendix B illustrates detailed corpus preprocessing, splits, and construction parameters.

Additional transfer evaluations. A paired construction set holds the source text fixed and compares the controlled mixer with an independent clock mixer on 240 further MultiNLI groups. Domain-transfer sets use WikiText, AG News, and speech-derived text without target-domain adaptation. AMI reuses the temporal track's 1,000-window view, and ICSI uses a separate 300-window reserve with no meetings shared with its 400-window development track. These evaluations, unitization rules, and the identifiability audit are specified in Appendix B.

Table 2: **Main results across five evaluation tracks on UNMIXBENCH.** Each block reports MP-WER (MP), permutation-aligned ownership accuracy (Own.), and evidence preservation (Ev.). Turn-marker ownership uses lexical alignment (Appendix C.2). Lower MP-WER is better; higher values are better for the other metrics. Best and second-best results are bold and underlined. Turn-marker uncertainty is given in Table 12, and Appendix D.5 reports hosted-model settings.

	MNLI			AMI			ICSI-dev			RB			DIG		
System	**MP↓**	**Own.↑**	**Ev.↑**	**MP↓**	**Own.↑**	**Ev.↑**	**MP↓**	**Own.↑**	**Ev.↑**	**MP↓**	**Own.↑**	**Ev.↑**	**MP↓**	**Own.↑**	**Ev.↑**
Generative baselines															
Compact source-array generation (4B)	0.6696	–	0.0845	.7275	–	.0423	.7439	–	.0400	.8415	–	.0240	.7310	–	.0464
Evidence-explicit source-array generation (4B)	0.6747	–	0.0814	.7418	–	.0343	.7530	–	.0317	.8547	–	.0272	.7314	–	.0544
Turn-marker generation (4B)	0.4271	0.6991	0.3957	0.6928	0.4441	0.6377	0.6690	0.4805	0.6600	0.6862	0.4959	0.3325	0.6449	0.5426	0.7769
Structured baselines															
Global Transformer	0.5368	0.6658	**1.0000**	.6406	.6086	**1.0000**	.6389	.6075	**1.0000**	.6820	.5866	**1.0000**	.7321	.5397	**1.0000**
EMA prototype	0.5165	0.6703	**1.0000**	.6596	.6201	**1.0000**	.6674	.6136	**1.0000**	.7427	.5814	**1.0000**	.7078	.5482	**1.0000**
RGS-CRF	0.4799	0.6917	**1.0000**	.6483	.6108	**1.0000**	.6495	.6076	**1.0000**	.6555	.5989	**1.0000**	.7032	.5552	**1.0000**
Legal projection	0.4788	0.6904	**1.0000**	.6431	**.6255**	**1.0000**	.6483	.6211	**1.0000**	.6843	.6015	**1.0000**	.6764	.5712	**1.0000**
Zero-shot frontier LLMs															
Qwen3.8-Flash (125B)	0.5420	–	0.4845	0.6546	–	0.6010	0.6486	–	0.5825	0.5984	–	0.6531	0.7674	–	0.2700
Qwen3.8-Max (2.4T)	0.3836	–	0.6803	**0.5975**	–	0.2330	**0.6056**	–	0.2525	**0.5428**	–	0.6759	**0.6171**	–	0.2483
DeepSeek-V4-Flash (284B)	0.8477	–	0.3742	0.9937	–	0.3450	1.0595	–	0.3725	0.9530	–	0.6074	1.1583	–	0.3300
DeepSeek-V4-Pro (1.6T)	0.6828	–	0.8000	0.7598	–	0.6040	0.7439	–	0.6175	0.6535	–	0.7863	0.8354	–	0.5583
DeepSeek-V4.1-Flash (552B)	0.5225	–	0.8091	0.6910	–	0.6850	0.6759	–	0.6575	0.6339	–	0.8127	0.8068	–	0.5167
Proposed method															
EPOR (4B; ours)	**0.3757**	**0.7457**	**1.0000**	.6320	.6205	**1.0000**	.6161	**.6228**	**1.0000**	.6156	**.6224**	**1.0000**	.6474	**.5912**	**1.0000**

6 EXPERIMENTS

EPOR is systematically evaluated on UNMIXBENCH. We compare it with free-generation and structured baselines across five principal tracks, then test the roles of canonical representation, ownership-route history, and decoding constraints. We also examine generalization across constructions, domains, and backbones, followed by an analysis of text-unmixing difficulty. Appendix D reports additional diagnostics and transfer results.

6.1 EXPERIMENTAL SETUP

Comparisons and baselines. Task-adapted systems share the Qwen3.5-4B backbone and SNLI data (Qwen Team, 2026). The free-generation baselines produce source-word arrays in two variants: compact source-array generation and evidence-explicit source-array generation. A separate turn-marker generator reproduces the mixed stream with source turn markers. Structured baselines—Global Transformer, EMA prototype, RGS-CRF, and legal projection—predict or recover indexed ownership. Frontier zero-shot models use evidence-explicit source generation. Ablations remove task adaptation, ownership-route history, or legality masking. PI-Route tests permutation-invariant supervision in place of canonical labels. Appendix A details these comparisons.

Training and inference. Task adaptation uses rank-32 NF4 QLoRA for two epochs and three runs (seeds 42–44) (Dettmers et al., 2023). EPOR uses teacher-forced route supervision and completion-safe beam search with $B = 4$. No target-track examples are used for training or checkpoint selection. Frontier calls use non-thinking inference at temperature zero (Appendix D.5). Outputs receive no post-hoc repair except in the designated legal-projection baseline.

Metrics. We use minimum-permutation WER (MP-WER), permutation-aligned ownership accuracy and switch F1 for indexed routes, and exact word-multiset preservation (Evidence). Structural validity follows Section 3, and route validity additionally requires canonical labels. Results are macro-averaged, with paired uncertainty grouped by provenance (Appendix C.2).

6.2 MAIN COMPARISON: OWNERSHIP PREDICTION VERSUS EVIDENCE REGENERATION

Among task-adapted systems, EPOR has the lowest mean MP-WER on four of five tracks and the lowest five-track average (see Table 2). On MultiNLI, it reduces MP-WER from 0.6696 to 0.3757 relative to compact source-array generation, a 43.9% relative reduction. Evidence-explicit instructions do not close this gap; the low Evidence scores reflect imperfect lexical reproduction (Appendix D.3).

Table 3: **EPOR component and representation ablations on MultiNLI.** Adapt. denotes task adaptation; Legal mask denotes completion-safe decoding. PI-Route is a matched representation control with noncanonical legal support, rather than a component-removal condition.

Condition	Adapt.	Legal mask	MP-WER ↓	Evidence ↑	Complete K-way ↑	Ownership acc. ↑	Switch F1 ↑
EPoR	✓	✓	0.3757	1.0000	1.0000	0.7457	0.8514
w/o task adaptation	×	✓	0.6842	1.0000	1.0000	0.5278	0.5333
w/o legality masking	✓	×	0.3819	1.0000	0.9721	0.7429	0.8492
w/o adaptation and legality masking	×	×	1.2008	1.0000	0.1473	0.3830	0.0894
Route-independent fixed-template history	✓	✓	0.4486	1.0000	1.0000	0.7113	0.7603
Representation control							
PI-Route	✓	✓	0.3782	1.0000	1.0000	0.7438	0.8521
Interface baseline							
Evidence-explicit source-array generation	✓	×	0.6747	0.0814	–	–	–

Generating words in mixed-stream order narrows the gap. Turn-marker generation reaches 0.4271 MP-WER on MultiNLI, but EPoR remains better on MultiNLI, AMI, ICSI, and ReadingBank, with all four paired 95% intervals excluding zero. On Digital, turn-marker generation is slightly lower (0.6449 versus 0.6474); the paired EPoR-minus-turn-marker difference is $+0.0026$ (95% CI $[-0.0067, 0.0120]$). EPoR has higher ownership accuracy on all five tracks and preserves evidence exactly, whereas turn-marker Evidence ranges from 0.3325 to 0.7769. Mixed-order generation therefore reduces reordering demands but still introduces lexical errors (Appendix D.4).

All structured baselines preserve evidence exactly, and their errors concern source ownership. EPoR reaches 0.7457 ownership accuracy on MultiNLI, though legal projection is more accurate on AMI. The 4B EPoR system is competitive with frontier zero-shot models in reconstruction, with the best MP-WER varying by track. Appendix D.5 reports their configurations and failure diagnostics.

6.3 Ablation and Component Analysis

Canonical representation. The matched PI-Route control replaces canonical supervision with exhaustive permutation-invariant training. It obtains 0.3782 MP-WER versus 0.3757 for EPoR (Table 3), with paired recovery intervals spanning zero. Its training requires 10.04 times as many LLM sequence evaluations and 4.08 times the wall time in the measured implementation. Canonicalization thus removes permutation redundancy at substantially lower training cost, with comparable recovery quality. Appendix D.7 gives the full comparison and cost accounting.

Assignment-specific history. Replacing ownership-route prefixes with a route-independent fixed template during training and evaluation raises MP-WER from 0.3757 to 0.4486 and lowers ownership accuracy from 0.7457 to 0.7113. Evidence and complete K-way structure remain unchanged, where legal actions still depend on each candidate's actual prefix, isolating the contribution of model-visible assignment history. Appendix D.6 also reports a graded history-corruption analysis.

Legality constraints and task adaptation. Removing the full legality mask after adaptation changes MP-WER from 0.3757 to 0.3819 and reduces complete K-way outputs from 1.0000 to 0.9721. The mask chiefly guarantees structure. With the same completion-safe decoder, task adaptation reduces MP-WER from 0.6842 to 0.3757.

Finite beam search. Increasing beam width from 1 to 32 lowers selected-route MP-WER from 0.3916 to 0.3737. This gold-aware diagnostic, unavailable at inference, reveals substantial ranking error even among legal candidates (Appendix D.12).

6.4 Generalization and Transfer

Ownership prediction outperforms source-array generation under an independent clock-based construction for Qwen3.5-4B, Qwen3.5-9B, and Gemma-7B-IT. These cross-backbone checks use

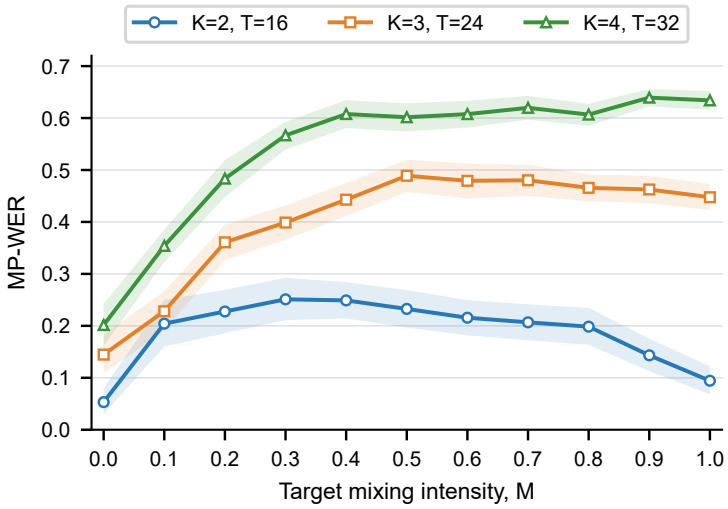


Figure 3: **Recovery across controlled mixing intensities.** The same source groups are evaluated at eleven target values of M. Points average three training runs and 80 source groups per (K, T) condition; shaded regions show source-group bootstrap 95% confidence intervals.

one training run per backbone. In domain transfer, EPoR improves over RGS-CRF on WikiText, the shared AMI view, and the separate ICSI reserve, but performs worse on AG News. Appendix D.9–D.11 gives construction, domain, and per-view serialization results.

6.5 What Makes Text Unmixing Difficult?

Our controlled analyses point to two distinct sources of difficulty: ambiguity in the ownership route and similarity between the latent sources. Route ambiguity is not monotonic in mixing intensity, as shown in Figure 3. With two sources, maximal switching forces an alternating route and can therefore make the ownership structure easier to recover, whereas intermediate mixing permits a less constrained set of assignments. Recovery is also worse for source groups with higher frozen-encoder similarity after exact matching on K, T, and target M, consistent with reduced lexical distinctiveness between sources. Appendix D.8 provides the corresponding controlled analyses.

7 Conclusion and Limitations

We introduce word-level text unmixing, a fine-grained de-interleaving problem in which complete lexical evidence is observed but source ownership is unknown. EPoR approaches this by separating history-conditioned ownership prediction from deterministic sequence reconstruction: it predicts canonical ownership routes under completion-safe constraints and then reconstructs each source directly from the indexed input occurrences. Across the five principal UnMixBench tracks, EPoR achieves the lowest mean MP-WER among task-adapted free-generation and structured baselines while preserving every observed word occurrence exactly, and remains competitive with zero-shot frontier LLMs. These results suggest that, when lexical evidence is fully observed, decoupling ownership inference from lexical regeneration provides a reliable alternative to direct source-text regeneration.

There are several limitations remaining in this work. First, the current formulation assumes that the source count K is known. Jointly inferring the number of sources and their ownership is an important extension. Second, finite-beam decoding guarantees structural validity but not correct ownership, and our candidate-ranking analysis indicates that substantial ranking error remains even among legal routes. Improving route scoring or reranking may therefore yield further gains. Third, our formulation isolates ownership recovery under complete lexical observations. Real pipelines may additionally contain ASR/OCR insertions, deletions, or substitutions, so extending evidence-preserving routing to noisy or incomplete observations remains an open problem.

Ethics Statement

We use public text and AMI/ICSI annotations and collect no new human-subject data; consequential source-attribution claims require independent evidence and human review.

Reproducibility Statement

The appendix and supplement specify the methods and provide the code, tests, configurations, and commands needed to reconstruct the datasets and rerun the reported experiments. An anonymized code snapshot is included with the submission.

AI Use Statement

Generative AI is used to improve manuscript readability and in the model-assisted first stage of the semantic-identifiability audit described in Appendix B.4. Specifically, GPT-5.6 Sol screened 120 examples for materially different legal partitions that could remain linguistically plausible; a human reviewer then examined all 120 cases and made the final judgments. The authors reviewed all AI-assisted work and remain responsible for the content and reported conclusions.

References

Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995. doi: 10.1162/neco.1995.7.6.1129.

Jean Berstel, Luc Boasson, Olivier Carton, Jean-Éric Pin, and Antonio Restivo. The expressive power of the shuffle product. *Information and Computation*, 208(11):1258–1272, 2010. doi: 10.1016/j.ic.2010.06.002.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*, pp. 632–642. Association for Computational Linguistics, 2015. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075/.

Sam Buss and Michael Soltys. Unshuffling a square is NP-hard. *Journal of Computer and System Sciences*, 80(4):766–776, 2014. doi: 10.1016/j.jcss.2013.11.002.

Jean-François Cardoso. Blind signal separation: Statistical principles. *Proceedings of the IEEE*, 86 (10):2009–2025, 1998. doi: 10.1109/5.720250.

Kent K. Chang, Danica Chen, and David Bamman. Dramatic conversation disentanglement. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4020–4046, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.248. URL https://aclanthology.org/2023.findings-acl.248/.

Zhuo Chen, Yi Luo, and Nima Mesgarani. Deep attractor network for single-microphone speaker separation. In *Proceedings of ICASSP*, pp. 246–250. IEEE, 2017. doi: 10.1109/ICASSP.2017.7952155.

Ta-Chung Chi and Alexander Rudnicky. Zero-shot dialogue disentanglement by self-supervised entangled response selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4897–4902. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.400. URL https://aclanthology.org/2021.emnlp-main.400/.

Christian Clausner, Stefan Pletschacher, and Apostolos Antonacopoulos. The significance of reading order in document recognition and its evaluation. In *2013 12th International Conference on Document Analysis and Recognition*, pp. 688–692. IEEE, 2013. doi: 10.1109/ICDAR.2013.141.

Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994. doi: 10.1016/0165-1684(94)90029-9.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0441. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1feb87871436031bdc0f2beaa62a049b-Abstract-Conference.html.

Micha Elsner and Eugene Charniak. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pp. 834–842. Association for Computational Linguistics, 2008. URL https://aclanthology.org/P08-1095/.

Micha Elsner and Eugene Charniak. Disentangling chat. *Computational Linguistics*, 36(3):389–409, 2010. URL https://aclanthology.org/J10-3004/.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-constrained decoding for structured NLP tasks without finetuning. In *Proceedings of EMNLP*, pp. 10932–10952. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.674. URL https://aclanthology.org/2023.emnlp-main.674/.

Sai R. Gouravajhala, Andrew M. Vernier, Yiming Shi, Zihan Li, Mark S. Ackerman, and Jonathan K. Kummerfeld. Chat disentanglement: Data for new domains and methods for more accurate annotation. In *Proceedings of the 21st Annual Workshop of the Australasian Language Technology Association*, pp. 112–117. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.alta-1.12/.

Chris Hokamp and Qun Liu. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of ACL*, pp. 1535–1546. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1141. URL https://aclanthology.org/P17-1141/.

Shota Horiguchi, Yusuke Fujita, Shinji Watanabe, Yawen Xue, and Kenji Nagamatsu. End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors. In *Proceedings of Interspeech*, pp. 269–273, 2020. doi: 10.21437/Interspeech.2020-1022. URL https://www.isca-archive.org/interspeech_2020/horiguchi20_interspeech.html.

Chengyu Huang, Zheng Zhang, Hao Fei, and Lizi Liao. Conversation disentanglement with bi-level contrastive learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2985–2996. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-emnlp.217. URL https://aclanthology.org/2022.findings-emnlp.217/.

Aapo Hyvärinen and Erkki Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4–5):411–430, 2000. doi: 10.1016/S0893-6080(00)00026-5.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. The ICSI meeting corpus. In *Proceedings of ICASSP*, pp. 364–367. IEEE, 2003. doi: 10.1109/ICASSP.2003.1198793.

Jyun-Yu Jiang, Francine Chen, Yan-Ying Chen, and Wei Wang. Learning to disentangle interleaved conversational threads with a Siamese hierarchical network and similarity ranking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1812–1822, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1164. URL https://aclanthology.org/N18-1164/.

Christian Jutten and Jeanny Hérault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991. doi: 10.1016/0165-1684(91)90079-X.

Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph J. Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C. Polymenakos, and Walter Lasecki. A large-scale corpus for conversation disentanglement. In *Proceedings of ACL*, pp. 3846–3856. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1374. URL https://aclanthology.org/P19-1374/.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 282–289. Morgan Kaufmann, 2001.

Yuke Lin, Ming Cheng, Ze Li, Beilong Tang, and Ming Li. Diarization-aware multi-speaker automatic speech recognition via large language models, 2025. URL https://arxiv.org/abs/2506.05796.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5755–5772, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.395. URL https://aclanthology.org/2022.acl-long.395/.

Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. Structural characterization for dialogue disentanglement. In *Proceedings of ACL*, pp. 285–297. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.23. URL https://aclanthology.org/2022.acl-long.23/.

Anthony Mansfield. An algorithm for a merge recognition problem. *Discrete Applied Mathematics*, 4(3):193–197, 1982. doi: 10.1016/0166-218X(82)90039-7.

Anthony Mansfield. On the computational complexity of a merge recognition problem. *Discrete Applied Mathematics*, 5(1):119–122, 1983. doi: 10.1016/0166-218X(83)90021-5.

Iain McCowan, Jean Carletta, Wessel Kraaij, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Wilfried Post, Dennis Reidsma, and Pierre Wellner. The AMI meeting corpus. In *Proceedings of Measuring Behavior*, pp. 137–140, 2005. URL https://groups.inf.ed.ac.uk/ami/corpus/.

Shikib Mehri and Giuseppe Carenini. Chat disentanglement: Identifying semantic reply relationships with random forests and recurrent neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 615–623, Taipei, Taiwan, 2017. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I17-1062/.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. URL https://arxiv.org/abs/1609.07843.

Stephen C. Milne. Restricted growth functions, rank row matchings of partition lattices, and q-Stirling numbers. *Advances in Mathematics*, 43(2):173–196, 1982. doi: 10.1016/0001-8708(82)90032-9.

Ari Pakman, Yueqi Wang, Catalin Mitelut, Jinhyung Lee, and Liam Paninski. Neural clustering processes. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7455–7465, 2020. URL https://proceedings.mlr.press/v119/pakman20a.html.

Tae Jin Park and Panayiotis Georgiou. Multimodal speaker segmentation and diarization using lexical and acoustic cues via sequence to sequence neural networks. In *Interspeech 2018*, pp. 1373–1377, 2018. doi: 10.21437/Interspeech.2018-1364. URL https://www.isca-archive.org/interspeech_2018/park18b_interspeech.html.

Tae Jin Park, Kunal Dhawan, Nithin Rao Koluguri, and Jagadeesh Balam. Enhancing speaker diarization with large language models: A contextual beam search approach. In *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 10861–10865, 2024. doi: 10.1109/ICASSP48485.2024.10446204. URL https://arxiv.org/abs/2309.05248.

Rohit Paturi, Sundararajan Srinivasan, and Xiang Li. Lexical speaker error correction: Leveraging language models for speaker diarization error correction. In *Interspeech 2023*, pp. 3567–3571, 2023. doi: 10.21437/Interspeech.2023-1982. URL https://www.isca-archive.org/interspeech_2023/paturi23_interspeech.html.

Qiming Peng, Yinxu Pan, Wenjin Wang, Bin Luo, Zhenyu Zhang, Zhengjie Huang, Yuhui Cao, Weichong Yin, Yongfeng Chen, Yin Zhang, Shikun Feng, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE-layout: Layout knowledge enhanced pre-training for visually-rich document understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3744–3756, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.274. URL https://aclanthology.org/2022.findings-emnlp.274/.

Matt Post and David Vilar. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of NAACL-HLT*, pp. 1314–1324. Association for Computational Linguistics, 2018. doi: 10.18653/v1/N18-1119. URL https://aclanthology.org/N18-1119/.

Qwen Team. Qwen3.5-4B model card. Hugging Face model card, 2026. URL https://huggingface.co/Qwen/Qwen3.5-4B. Accessed 2026-08-25.

Antonio Restivo. The shuffle product: New research directions. In *Language and Automata Theory and Applications*, volume 8977 of *Lecture Notes in Computer Science*, pp. 70–81. Springer, 2015. doi: 10.1007/978-3-319-15579-1_5.

Romeo Rizzi and Stéphane Vialette. On recognizing words that are squares for the shuffle product. In *Computer Science – Theory and Applications*, volume 7913 of *Lecture Notes in Computer Science*, pp. 235–245. Springer, 2013. doi: 10.1007/978-3-642-38536-0_21.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. GoLLIE: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Y3wpuxd7u9.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of EMNLP*, pp. 9895–9901. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.779. URL https://aclanthology.org/2021.emnlp-main.779/.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4077–4087. Curran Associates, Inc., 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, 1974. doi: 10.1145/321796.321811.

Junlin Wang, Jue Wang, Zhen Xu, Ben Athiwaratkun, Bhuwan Dhingra, Ce Zhang, and James Zou. Staircase streaming for low-latency multi-agent inference, 2025. URL https://arxiv.org/abs/2510.05059.

Quan Wang, Yiling Huang, Guanlong Zhao, Evan Clark, Wei Xia, and Hank Liao. DiarizationLM: Speaker diarization post-processing with large language models. In *Interspeech 2024*, pp. 3754–3758, 2024. doi: 10.21437/Interspeech.2024-209. URL https://www.isca-archive.org/interspeech_2024/wang24h_interspeech.html.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*, 2023.

Zilong Wang, Yiheng Xu, Lei Cui, Jingbo Shang, and Furu Wei. LayoutReader: Pre-training of text and layout for reading order detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4735–4744, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.389. URL https://aclanthology.org/2021.emnlp-main.389/.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101/.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *Proceedings of the First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=BAakY1hNKS.

Han Yin, Yafeng Chen, Chong Deng, Luyao Cheng, Hui Wang, Chao-Hong Tan, Qian Chen, Wen Wang, and Xiangang Li. SpeakerLM: End-to-end versatile speaker diarization and recognition with multimodal large language models, 2025. URL https://arxiv.org/abs/2508.06372.

Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *Proceedings of ICASSP*, pp. 241–245. IEEE, 2017. doi: 10.1109/ICASSP.2017.7952154. URL https://arxiv.org/abs/1607.00325.

Tao Yu and Shafiq Joty. Online conversation disentanglement with pointer networks. In *Proceedings of EMNLP*, pp. 6321–6330. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.512. URL https://aclanthology.org/2020.emnlp-main.512/.

Eva Zangerle, Maximilian Mayerl, Martin Potthast, and Benno Stein. Overview of the style change detection task at PAN 2022. In *Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum*, volume 3180 of *CEUR Workshop Proceedings*, pp. 2344–2356. CEUR-WS.org, 2022. URL https://ceur-ws.org/Vol-3180/paper-186.pdf.

Chong Zhang, Ya Guo, Yi Tu, Huan Chen, Jinyang Tang, Huijia Zhu, Qi Zhang, and Tao Gui. Reading order matters: Information extraction from visually-rich documents by token path prediction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13716–13730. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.846. URL https://aclanthology.org/2023.emnlp-main.846/.

Chong Zhang, Yi Tu, Yixi Zhao, Chenshu Yuan, Huan Chen, Yue Zhang, Mingxu Chai, Ya Guo, Huijia Zhu, Qi Zhang, and Tao Gui. Modeling layout reading order as ordering relations for visually-rich document understanding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9658–9678, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.540. URL https://aclanthology.org/2024.emnlp-main.540/.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28*, pp. 649–657, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html.

Rongxin Zhu, Jey Han Lau, and Jianzhong Qi. Findings on conversation disentanglement. In *Proceedings of the 19th Annual Workshop of the Australasian Language Technology Association*, pp. 1–11. Australasian Language Technology Association, 2021. URL https://aclanthology.org/2021.alta-1.1/.

APPENDIX CONTENTS

A Experimental Details **16**

A.1 Baselines and Matched Controls 16

A.2 Training configuration 17

A.3 Prompts and output formats 17

B Dataset Construction **19**

B.1 Corpora and lexical units 19

B.2 Interleaving mechanisms 20

B.3 Speech-derived text windows 21

B.4 Semantic-identifiability audit 21

C Formal Details and Guarantees **22**

C.1 Valid Output Space and Permutation Invariance 22

C.2 Evaluation Metrics 22

C.3 Repeated-Word Alignment 23

C.4 Canonical Representation and Supervision 23

C.5 Completion-Safe Decoding 23

C.6 Exact Reconstruction Guarantee 24

C.7 Training and Inference Algorithms 24

D Extended Experiments **24**

D.1 Ownership-interface ablations and diagnostics 24

D.2 Full MultiNLI metrics 25

D.3 Generation Baselines: Prompt Variants and Error Diagnostics 26

D.4 Task-matched turn-marker generation 26

D.5 Frontier zero-shot language models 27

D.6 Ownership adaptation and history 27

D.7 Permutation-invariant representation comparison 28

D.8 Difficulty factors 29

D.9 Construction and backbone details 30

D.10 Domain transfer 31

D.11 Serialization transfer 31

D.12 Candidate quality and ranking 31

E Extended Related Work **32**

E.1 Shuffle Languages and String Unshuffling 32

E.2 Reading-Order Recovery in Document Understanding 33

E.3 Conceptual Relation to Blind Source Separation 33

Table 4: Implementation summary and trainable parameter counts for the comparison baselines.

Learning and model state

System	Scoring / state	Trainable head	Loss
Free generation	causal source-text likelihood	none beyond QLoRA	assistant-token LM
Turn-marker generation	causal marked-stream likelihood	none beyond QLoRA	assistant-token LM
Global Transformer	bidirectional contextual emissions	256-d projection; 2-layer, 8-head encoder; linear emission	occurrence CE
EMA prototype	emission plus cosine to per-source EMA ($\lambda = .75$, scale 4)	256-d projection/query; shared biases	occurrence CE
RGS-CRF	global emissions plus 4×4 transitions	Global head plus transition matrix	exact legal-route CRF NLL
Legal projection	edit cost to generated source arrays; EMA tie guide	none in projection	none

Inference and trainable size

System	Decoder / projection	Trainable parameters
Free generation	greedy JSON generation	64,929,792
Turn-marker generation	greedy word-and-marker generation	64,929,792
Global Transformer	legal RGS beam, $B = 4$	67,166,468
EMA prototype	stateful legal RGS beam, $B = 4$	65,652,486
RGS-CRF	emission/transition RGS beam, $B = 4$	67,166,484
Legal projection	exact shuffle DP; otherwise anchor beam 512	0 (projection)[†]

[†]Projection adds no trainable parameters to its generator and guide checkpoints.

A Experimental Details

A.1 Baselines and Matched Controls

Comparison baselines. Table 4 summarizes the components required to reproduce each baseline comparison. The task-finetuned generator predicts K source-word arrays in nested JSON and is trained with assistant-token causal language-modeling loss. The task-specific baseline routing heads use the last LM subword state aligned to each input word; EPoR instead uses the existing vocabulary head.

The task-matched turn-marker generator (Park & Georgiou, 2018) uses the same backbone, training examples, adaptation family, and generation budget as source-array generation. Its target retains the mixed-stream order, copies every word occurrence, and inserts <S0>, <S1>, and subsequent canonical markers before the first word and whenever ownership changes. The first marker is <S0>, new markers follow first-appearance order, and all K markers must occur. A strict full-output parser rejects explanations, malformed or redundant markers, missing sources, and non-word-layer items. The training context is 1,152 LM tokens because two complete training targets exceed 1,024; this avoids dropping or truncating examples. All other data, optimization, seed, and evaluation choices follow the matched source-generation protocol.

Legal projection first accepts a proposal when an exact shuffle-certificate DP verifies it (state limit 250,000). Otherwise, a beam of 512 maximizes ordered word anchors, charges unmatched observed and skipped generated words as edit cost, and uses the aligned EMA route only to break edit ties; every result is then reconstructed by stable filtering.

The structured baselines draw on self-attention (Vaswani et al., 2017), prototype classification (Snell et al., 2017), conditional random fields (Lafferty et al., 2001), and edit-distance projection (Wagner & Fischer, 1974). These citations identify component families; the task-specific implementations in Table 4 use different ownership scores or projection objectives.

Matched representation control. PI-Route is a matched representation control for first-appearance canonicalization, using exhaustive permutation-invariant training (Yu et al., 2017). It replaces canonical targets with arbitrary source labels and enumerates all $K!$ global relabelings for the minimum LM-token-normalized negative log-likelihood. It matches EPoR in backbone, trainable parameters, data, learning rate, epochs, context length, source counts, training runs, beam budget, reconstruction map, and metrics.

A.2 Training configuration

The task-adapted comparisons use post-trained Qwen3.5-4B with NF4 quantization, rank-32 QLoRA, $\alpha = 64$, dropout 0.05, two epochs, effective batch size 16, and three training runs (Dettmers et al., 2023). EPoR and PI-Route use learning rate 2×10^{-4} and a context of 1,024 LM tokens. The Global, EMA, and RGS-CRF routers use 10^{-4} and a context of 512 LM tokens. A length audit finds that 3.4% of the supervised EPoR sequences exceed 512 LM tokens and none exceeds 1,024.

The cross-backbone experiments use one training run per backbone for Qwen/Qwen3.5-4B, Qwen/Qwen3.5-9B, and google/gemma-7b-it. They keep the 10,000 training mixtures, two epochs, rank-32 QLoRA, a context of 1,024 LM tokens, effective batch size 16, known-K interface, and evaluation examples fixed. Batch size and gradient accumulation are adjusted to preserve the effective batch. Each backbone passes a synthetic known-route test, atomic-symbol checks, exact mask checks, stable reconstruction, non-emptiness, and word-layer evaluation before the reported evaluation.

Hardware. The EPOR–PI-Route runtime comparison used the same NVIDIA GeForce RTX 5090 (32 GB) host for all six training runs (three seeds per method). Appendix D.7 specifies the timing scope.

A.3 Prompts and output formats

This section gives the prompts, output formats, and parsing rules used in the experiments. Input words are indexed and JSON-encoded before applying the model-specific chat template. The panels below distinguish system instructions, instance templates, substituted fields, and expected responses visually without altering the prompt text. In these verbatim prompts, input "tokens" denote task-level word occurrences.

EPOR prompt template

System message

```
You perform source routing under an external grammar.
Emit exactly one single-digit
source label per indexed input token, with no spaces or punctuation.
The decoder enforces legal labels.  Do not emit any explanation.
```

User template

```
The sequence contains {K} latent sources.  Source order is arbitrary.
Emit exactly {T} digits, one label in [0, {K_minus_1}] per token.
The
first label must be 0.  A new label may only be introduced as one plus
the largest previous label.  Every source must occur at least once.
index\ttoken
{indexed_json_tokens}
```

Expected response

```
Compact labels:  012001...
```

EPOR prompt template. Color distinguishes the system instruction, instance template, and expected response; blue text marks substituted fields.

The displayed template is the decimal-codec prompt. Its target is the concatenated zero-based route without separators, corresponding to the one-based mathematical labels in Section 4. The prompt is loss-masked; route symbols and the assistant termination suffix are supervised. At inference, the decoder emits exactly T ownership symbols, excludes EOS from the route alphabet, and rejects unsupported output tokens.

Compact source-array generation prompt template

System message

```
You
reconstruct latent source texts from an interleaved token sequence.
Return exactly one JSON array per
source, where every source is represented as an array of word tokens.
Source order
is arbitrary.  Return only one nested JSON array and no explanation.
```

User template

```
The observed sequence interleaves
{K} latent sources while preserving word order inside each source.
Reconstruct {K} source token arrays from the {T} observed tokens.
index\ttoken
{indexed_json_tokens}
```

Expected response

```
JSON source token arrays:  [["word",...],["word",...]]
```

Compact source-array generation prompt template. The formatting separates instructions supplied to the model from instance fields and returned JSON.

The parser selects the last nested JSON value containing exactly K arrays of strings. Parse failure is scored as an invalid or empty prediction; no legal-route repair is applied to the free-generation output. The comparison measures this task-finetuned generator–parser pipeline.

Evidence-explicit source-array generation. The second task-finetuned Qwen comparison and the DeepSeek-V4 and DeepSeek-V4.1-Flash evaluations use the prompt shown below.

Evidence-explicit source-array generation prompt template

System message

```
You
reconstruct latent source sequences by partitioning indexed observed
word occurrences.  Return exactly one non-empty JSON array per
source, where every element is a word string copied from the input.
Use every observed
word occurrence exactly once.  Do not add, delete, duplicate,
correct, merge, split, or paraphrase words.  Preserve increasing
input-index order within every source.  Source order is arbitrary.
Return only one valid nested JSON array and no explanation.
```

User template

```
The observed
sequence contains {T} indexed word occurrences from {K} latent
sources.  Partition all occurrences into exactly {K} non-empty source
token arrays.  Copy each token string exactly once across the arrays.
Do not add, delete, duplicate, correct, merge, split, or paraphrase
any token.  Preserve increasing input-index order within each source.
index\ttoken
{indexed_json_tokens}
```

Expected response

```
JSON source token arrays:  [["word",...],["word",...]]
```

Evidence-explicit source-array generation prompt template used for the task-finetuned comparison and hosted zero-shot models.

The task-finetuned Qwen system uses the same data, adaptation budget, three training runs, decoding budget, parser, and evaluation examples as the compact generator. Hosted evaluations use the same

evidence-explicit prompt and parser in non-thinking mode. Calls were issued through the DeepSeek API or Alibaba Cloud Model Studio under the provider-facing aliases available at evaluation time. For all generators, invalid JSON or the wrong number of arrays is an empty prediction, and no repair is applied.

B Dataset Construction

B.1 Corpora and lexical units

Tables 5–7 summarize corpus-specific unitization, split construction, and serialization-track statistics, respectively. The corpora are (Bowman et al., 2015; Williams et al., 2018; Merity et al., 2016; Zhang et al., 2015; McCowan et al., 2005; Janin et al., 2003; Wang et al., 2021). Repeated strings are not deduplicated inside an example; each word occurrence retains its input position. A word occurrence is a task-level unit produced by the corpus-specific unitizer, including punctuation where retained. LM tokens refer to the model's subword encoding and may subdivide a word occurrence. In Tables 5 and 6, "Examples / groups" lists materialized examples first and provenance groups second; different renderings of the same group count as separate examples.

Operational ReadingBank sources. For each page, the observed sequence is the left-to-right/top-to-bottom page serialization, indexed $0, \ldots, n-1$. ReadingBank also supplies a permutation $q = (q_1, \ldots, q_n)$ such that selecting observed positions in that order recovers the DOCX-XML reading-order sequence. We split q immediately before each descent, $q_{j+1} < q_j$; the resulting maximal increasing runs define the reading flows. Each observed position receives the ID of the run that contains it and its rank within that run. These are deterministic, metadata-derived flows, not human-annotated semantic sources or layout regions. For example, $q = (0,3,1,4,2,5)$ yields runs $(0,3)$, $(1,4)$, and $(2,5)$ and observed-order labels $(0,1,2,0,1,2)$. Released sidecars make the construction traceable through `page_id`, `window_start`, `window_length`, `gold_route`, `source_local_indices`, and `decomposition_version`; for instance, `aqa-70372-sow-humang-globalsgoverng.docx:5` uses the 32-word window starting at observed position 112. Its first twelve observed positions have flow labels (0,0,0,0,1,1,2,2,2,3,3,3); the first six corresponding source-local ranks are (23,24,25,26,32,33) in the frozen sidecar.

Candidate windows have length 32 or 64 and stride 8 (including the final possible start). We retain windows with $K \in \{2,3,4\}$, at least eight words per active flow, at least K switches, and a valid one-to-one normalized word inventory. One eligible window per page is chosen by a result-blind hash with seed 20260915; a second deterministic ranking caps each (K,T) stratum at 150 examples. Documents used by the development view are excluded. This procedure yields 833 examples from 828 test documents.

The serialization extension contains 3,433 evaluation views. Its 1,200 digital views are three schedules over 400 shared source groups, so group-level inference keeps the schedules together. The ReadingBank view contains 833 examples from 828 test documents and excludes all documents used to construct its development view. Each track was filtered without model predictions and validated for non-empty sources, one-time occurrence use, source-local order, deterministic replay, and ownership-field isolation. We manually inspected ten examples per serialization mechanism to verify that the mixtures followed the intended serialization patterns.

The 10,000-example training set draws $K = (2,3,4)$ with probabilities $(0.6,0.3,0.1)$. The realized counts are 5,955, 2,998, and 1,047, with 25,092 source spans consumed without reuse. Source lengths vary in training. The controlled MultiNLI evaluation contains 240 source/document-disjoint groups, balanced over K and four bands of frozen-encoder source similarity. Each group is reused only across eleven target mixing intensities. The paired construction evaluation uses another 240 disjoint groups, with identical source word occurrences under both route generators. Source similarity S is the mean pairwise cosine after attention-masked mean pooling and L2 normalization with `sentence-transformers/all-MiniLM-L6-v2`. Empirical quartile cut points (within K) are $(0.0079,0.0659,0.1332)$ for $K=2$, $(0.0316,0.0674,0.1066)$ for $K=3$, and $(0.0476,0.0731,0.1029)$ for $K=4$.

Table 5: Corpora, roles, and preprocessing used in the experiments.

Corpus	Role	Examples / groups	Source unit and preprocessing
SNLI 1.0	Train	10,000 / 10,000	Premise and hypothesis spans; deduplication across held-out splits, NFKC normalization, whitespace collapse, and deterministic regex unitization.
SNLI 1.0	Development	1,000 / 1,000	Same preprocessing as training; source/text-disjoint from training and test data.
MultiNLI 1.0	Controlled evaluation	2,640 / 240	Premises only; NFKC normalization, whitespace collapse, and eight whitespace units per source. Prompt and document identifiers define provenance.
MultiNLI 1.0	Paired construction	480 / 240	Same fixed-length preprocessing; identical source word occurrences under controlled and independent-clock mixing.
MultiNLI 1.0	Unequal-length stress	540 / 180	Variable source spans; independent-clock, long-run, and bursty mixing. Prompt and document identifiers define provenance.
WikiText-103	Domain transfer	1,448 / 1,448	Length-filtered sentence spans materialized with the deterministic regex unitizer.
AG News	Domain transfer	2,500 / 2,500	Official-test article text filtered to 12–80 regex units.
AMI 1.6.2	Speech-domain and temporal-serialization transfer	1,000 / 59 meetings	NFKC-normalized XML word nodes, including punctuation; 64-unit windows.
ICSI core	Speech-domain transfer (consumed reserve)	300 / 15 meetings	NFKC-normalized word nodes with deterministic timing repair; 64-unit windows.
ICSI core	Temporal serialization (development)	400 / 15 meetings	A development view replayed from the frozen timestamp-merge construction; 64-unit windows.
ReadingBank	Spatial serialization	833 / 828 documents	XML-reading-order segments retained as deterministic metadata-derived reading flows; 32- or 64-unit windows.
WikiText-103	Digital serialization	1,200 / 400 groups	The same 400 source groups rendered with smooth, bursty, and unequal-rate virtual event schedules.

Table 6: Core UNMIXBENCH specification. All held-out groups are provenance-disjoint from training/development; the paired and stress sets are also disjoint from the controlled MultiNLI evaluation.

Split	Corpus	Source unitization	Examples / groups	K	T/length	Interleaving	Disjointness	Seed
Train	SNLI	NFKC + regex words	10,000 / 10,000	2–4	variable	controlled IID	held-out source/text	2027
Development	SNLI	NFKC + regex words	1,000 / 1,000	2–4	variable	controlled IID	train/test source/text	2027
Controlled evaluation	MultiNLI	first 8 whitespace units/source	2,640 / 240	2–4	16/24/32	11 controlled intensities	source + document	2701
Paired construction	MultiNLI	first 8 whitespace units/source	480 / 240	2–4	16/24/32	controlled + renewal clock	source + document	13013
Unequal-length stress	MultiNLI	variable source spans	540 / 180	2–4	variable	clock + long-run + bursty	source + document	20270905

B.2 INTERLEAVING MECHANISMS

Let $N_{\text{switch}} = \sum_{t=2}^{T} \mathbf{1}[z_t \neq z_{t-1}]$. For $T > K$, mixing intensity is

$$M = \frac{N_{\text{switch}} - (K-1)}{T-K} \in [0,1]. \tag{11}$$

The controlled mixer sets the target switch count to

$$N^*_{\text{switch}} = \text{round}\{(K-1) + M(T-K)\}. \tag{12}$$

It allocates positive runs across sources, arranges run labels without equal neighbors, divides each source into balanced positive run lengths, and consumes each source from left to right. Integer switch counts may realize the nearest rather than the requested M. The normalized measure is undefined for $T = K$. With unequal source lengths, length imbalance can prevent a switch at every position, so the largest attainable intensity may be below one.

Table 7: Serialization-transfer extension. Temporal and spatial tracks use real metadata; digital profiles simulate a concurrent-production mechanism with virtual event-time schedules and reuse the same source groups. Counts describe frozen evaluation views before model inference.

Track	Serialization mechanism	Examples	Groups	K counts	T counts	Mean M
AMI	annotated word-time merge	1,000	59	2:667; 3:333	64:1,000	0.275
ICSI dev	annotated/repaired word-time merge	400	15	2:200; 3:200	64:400	0.290
ReadingBank	metadata-derived reading flows	833	828	2:300; 3:300; 4:233	32:383; 64:450	0.176
Digital–smooth	virtual event-time merge	400	400	2:134; 3:133; 4:133	32:201; 64:199	0.582
Digital–bursty	virtual event-time merge	400	400	2:134; 3:133; 4:133	32:201; 64:199	0.266
Digital–unequal-rate	virtual event-time merge	400	400	2:134; 3:133; 4:133	32:201; 64:199	0.531

The independent-clock mixer is implemented separately. For each source k, it draws i.i.d. inter-arrivals $\Delta_{k,j} \sim \mathrm{Exponential}(1)$ and cumulative times $\tau_{k,j} = \sum_{r \le j} \Delta_{k,r}$. Events are merged by $(\tau_{k,j}, k, j)$, which preserves source-local order and provides a deterministic tie break.

Digital scheduling profiles. The digital track replays each of 400 WikiText source groups under three virtual-time schedules. In *smooth*, every source has independent inter-arrivals $\Delta \sim \mathrm{Exponential}(1)$. In *unequal-rate*, source indices $0, 1, 2, 3$ use rates $(1.0, 1.5, 0.7, 2.0)$, truncated to the realized K. In *bursty*, the first word of a burst follows a long-pause increment $\mathrm{Exponential}(0.22)$, the burst length is sampled uniformly from $\{2,3,4,5\}$ words, and subsequent words use $\mathrm{Exponential}(4.0)$ increments. Per-source increments are accumulated, and all word events are sorted by $(\text{event time}, \text{source index}, \text{source-local index})$. The global construction seed is 20260915; a deterministic group seed is derived for each source group and reused across profiles. The frozen implementation is identified by builder `digital_virtual_time_v1` and serializer `virtual_event_time_merge_v1`. The same lexical sources are used for all three profiles, so only the scheduling mechanism changes.

The unequal-length stress test adds two mechanisms. The long-run generator creates a small number of extended contiguous runs, while the bursty generator alternates short bursts with variable length. The 180 source groups include more than one total length for every $K \in \{2,3,4\}$. Every builder verifies that each source is non-empty, every indexed word occurrence is used once, source-local order is preserved, and repeated generation with the same input state is deterministic.

B.3 Speech-derived text windows

AMI words are read from speaker-specific XML and sorted by start time, end time, speaker identifier, and source-local index. ICSI uses an annotated start when available, otherwise an annotated end or deterministic channel-local interpolation; wholly untimed channels are excluded. These rules flatten overlapping annotations rather than remove them. Both builders scan 64-unit windows at stride 32, retain $K \in \{2,3\}$, require at least eight words per represented speaker, and supply only the flattened words and K to the model. The domain-transfer analysis uses the frozen 1,000-window AMI view from 59 meetings and a 300-window ICSI reserve from 15 reserve meetings. The temporal-serialization comparison reuses the same AMI examples and predictions, so the two AMI roles are not independent evaluations. For ICSI, it instead uses a separate frozen 400-window development view from 15 validation meetings. The ICSI development and reserve meeting sets are disjoint, and no example is shared between them. Intervals resample complete meetings.

B.4 Semantic-identifiability audit

We inspected a stratified panel of 120 controlled-MultiNLI examples spanning $K \in \{2,3,4\}$, four source-similarity bands, and five mixing settings. The GPT-5.6 Sol reviewer agent using the extra-high reasoning setting was shown the mixed text, K, and the recorded source partition and asked to construct a materially different legal partition that remained linguistically plausible. The screen found no clear alternative for 103 examples, marked seven as uncertain, and flagged three as plausible alternatives and seven as strongly underdetermined. A human reviewer then examined all 120 examples with access to the mixed text, K, and recorded source partition, using semantic, grammatical, and discourse coherence. The human reviewer found no materially different plausible alternative for the 103 examples initially marked clear. Of the 17 non-clear cases, only two $K = 4$ examples, at target $M = 0.75$ and

$M = 1.0$, retained a plausible alternative; the other 15 were judged to have one defensible partition. Thus, the two-stage model-assisted and human audit found no materially different plausible alternative in 118 examples, while two retained one. The stratified panel was designed to probe difficult cases, so this descriptive rate is neither an unbiased prevalence estimate nor a human ceiling on MP-WER.

C Formal Details and Guarantees

C.1 Valid Output Space and Permutation Invariance

Equation 1 defines all surjective labeled source assignments. The symmetric group S_K acts on this space by global relabeling, $(\pi \cdot z)_t = \pi(z_t)$. Because every label occurs, each assignment orbit has $K!$ members. Stable filtering maps globally relabeled assignments to source tuples that differ only by the same permutation. Therefore Equation 3 is a quotient output space: it contains non-empty, occurrence-conserving, order-preserving source tuples modulo global source permutation. This support follows from the task definition and does not depend on a learned score or a particular prediction interface. Repeated words can make different indexed assignments lexically indistinguishable (Appendix C.3), and different partitions can also remain semantically plausible.

C.2 Evaluation Metrics

For predicted sources $\widehat{Y}$ and gold sources Y, MP-WER is

$$\text{MP-WER}(\widehat{Y}, Y) = \min_{\pi} \frac{\sum_{k=1}^{K'} d_{\text{edit}}(\widehat{y}_k, y_{\pi(k)})}{\max(1, \sum_{y \in Y} |y|)}, \tag{13}$$

where $K' = \max(|\widehat{Y}|, |Y|)$ and the smaller side is padded with empty sources. For a predicted ownership route, ownership accuracy is

$$\text{OwnAcc}(\widehat{z}, z^\star) = \max_{\pi \in S_K} \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[\pi(\widehat{z}_t) = z_t^\star]. \tag{14}$$

Ownership accuracy and macro F1 use the source permutation that maximizes positional agreement. This alignment remains necessary with canonical labels: different predicted and gold partitions can introduce corresponding sources in different orders. Switch F1 compares the label-invariant boundary events $z_t \neq z_{t-1}$. Source-array free-generation baselines do not provide an indexed ownership route, so their ownership metrics are left undefined.

For turn-marker generation, we first align the generated mixed-order words to the input by minimum word edit distance, with unit insertion, deletion, and substitution costs. Among optimal paths, we maximize exact word matches, then trace forward with a fixed preference for matching, substitution, deletion, and insertion. This lexical alignment uses no ownership labels. Only exact word matches transfer marker labels to observed positions; deleted or substituted input words remain unassigned and count as errors. Inserted words receive no input position. Ownership accuracy uses all T input positions, with source-label permutation optimized after word alignment. A predicted switch requires matched labels on both adjacent input positions; every gold switch remains in the recall denominator. Strict parsing failures receive zero ownership scores. MP-WER and Evidence use the generated source texts. Thus, for turn-marker generation, ownership accuracy is an end-to-end measure over input positions that penalizes unmatched words; it is not a pure attribution score conditioned on successful lexical reproduction.

Word-multiset preservation (Evidence) is one for an example if every word string has the same total multiplicity in the reconstructed sources and the input, and zero otherwise. The reported mean is the fraction of examples satisfying this test; it is not a fraction of individually preserved words. Evidence alone checks neither ownership nor source-internal order and cannot distinguish repeated occurrences with the same surface form. Task-level structural validity requires K non-empty sources, exact occurrence use, and source-local order. For assignment-based systems, route validity additionally checks assignment length, label range, canonical first appearance, and use of every source. Recovery and support metrics are reported separately.

C.3 Repeated-Word Alignment

Repeated surface forms can admit several indexed alignments even when the gold source strings are fixed. A dynamic-programming state $c = (c_1,...,c_K)$ records the consumed prefix length of every source. At input position t, transition k is allowed exactly when $c_k < |y_k|$ and $x_t = y_{k,c_k+1}$. Counts of paths reaching the same state are summed, and the terminal count is the number of indexed-source alignment paths. Source indices remain fixed inside this computation. Thus $x = (a,a)$ with two identical singleton sources has two indexed paths but only one unlabeled partition. We use this count only as an ambiguity diagnostic; MP-WER operates on the reconstructed source strings.

C.4 Canonical Representation and Supervision

For any partition of the T input positions into K non-empty unlabeled blocks, order the blocks by their smallest position and label them $1,...,K$. The resulting route satisfies Equation 4. Conversely, forgetting the labels of a route in $\mathcal{Z}_{T,K}$ recovers the indexed partition. These maps are inverse. First-appearance canonicalization therefore selects one member from each orbit of the task-level assignment space, giving

$$\mathcal{P}^{\mathrm{lab}}_{T,K}/S_K \cong \mathcal{Z}_{T,K}. \tag{15}$$

There are $K!S(T,K)$ labeled surjections and $S(T,K)$ canonical routes, where $S(T,K)$ is the Stirling number of the second kind. This correspondence is between canonical routes and indexed partitions, not between routes and unique source strings. Canonicalization removes label redundancy but does not resolve semantic ambiguity.

Ownership encoding scheme and notation. Each word occurrence is assigned one output LM token, while its indexed input representation may contain several LM tokens. The encoding scheme checks that ownership symbols have distinct token IDs and encode atomically both alone and in adjacent sequences. Decimal symbols are used when these checks pass; otherwise, verified existing reserved tokens provide the route alphabet. No vocabulary entries or ownership head are added. Unsupported or non-atomic encodings fail validation. Mathematical routes use labels $1,...,K$; the implementation and decimal prompts use $\widetilde{z}_t = z_t - 1 \in \{0,...,K-1\}$. This offset changes notation, not the partition or admissibility rule.

Supervision implementation. The indexed input and K are formatted with the model-specific chat template, followed by the serialized canonical route and the standard assistant termination suffix. Prompt and padding positions are excluded from the loss; route and termination positions are supervised. Algorithm 1 and Equation 7 specify the training procedure, and Appendix A.2 gives the QLoRA configuration.

C.5 Completion-Safe Decoding

Consider a canonical prefix before position t, with largest label m_{t-1} and $m_0 = 0$. The label range in Equation 8 is necessary and sufficient for preserving first appearance in the next step. After choosing a, there are $K - \max(m_{t-1},a)$ unseen sources and $T - t$ remaining positions. A completion exists if and only if the latter count is at least the former. Each missing source can be introduced once in order, after which surplus positions can be assigned to an existing source. Thus $\mathcal{A}_t$ contains exactly the actions that admit a complete canonical route. Starting from the empty prefix and applying this rule for T steps yields a member of $\mathcal{Z}_{T,K}$.

Implementation details. Algorithm 2 gives the complete decoding procedure. The admissibility mask is computed separately for each retained prefix and keeps the original score of every legal extension:

$$\ell_t(a \mid z_{<t}) = \begin{cases} \log p_\theta(a \mid x,K,z_{<t}), & a \in \mathcal{A}_t(z_{<t}), \\ -\infty, & \text{otherwise.} \end{cases}$$

This mask restricts the candidate set rather than redefining a normalized probability distribution over legal labels. EOS is excluded from the route alphabet, and decoding stops after exactly T ownership symbols. An empty beam or a route that fails the encoding, canonicality, non-emptiness, order, or evidence-preservation checks is rejected rather than repaired. Finite-beam search approximates maximization of

Equation 5; it does not enumerate the full support or guarantee a globally optimal route. Every accepted route is nevertheless a complete canonical partition. Unlike training, inference uses predicted prefixes and fixed-length termination; the supervised assistant suffix is not used to decide when to stop routing.

C.6 Exact Reconstruction Guarantee

The admissibility rule characterizes the full feasible route space, whereas a finite beam generally explores only part of it. Beam pruning cannot make a retained prefix invalid, but it can discard the globally highest-scoring route. Validity therefore holds for every positive beam width without an optimality guarantee. For a complete route, the sets $\{t : z_t = k\}$ are disjoint, cover $\{1,...,T\}$, and are non-empty. Reading each set in increasing index uses every input position once and preserves order within each assigned source. It therefore reproduces the materialized input-occurrence multiset without assuming correct ownership. These guarantees concern the task-level occurrences, not raw-file formatting. Unit tests enumerate routes for small (T,K) and compare Equation 8 with exhaustive completions.

C.7 Training and Inference Algorithms

Algorithms 1 and 2 give the complete training and inference procedures described in Section 4.

Algorithm 1 EPoR training

Require: Examples $\{(x_i, K_i, z_i^\star)\}$; causal LM p_θ

Canonicalize each $z_i^\star$ by first appearance
for all minibatches $\mathcal{M}$ **do**
  for all $(x, K, z^\star) \in \mathcal{M}$ **do**
    $P \leftarrow \textsc{FormatPrompt}(x, K)$
    $v^\star \leftarrow \textsc{SerializeRoute}(z^\star)$
    Append assistant termination suffix to $v^\star$
    Form teacher-forced input $[P; v^\star]$
    Mask prompt and padding from loss
  end for
  Compute $\mathcal{L}_{\text{route}}$ on $\mathcal{M}$ using Equation 7
  Update QLoRA parameters
end for
return adapted scorer p_θ

Algorithm 2 EPoR completion-safe inference

Require: $x = (x_1, \ldots, x_T)$; K; scorer p_θ; beam width B

$\mathcal{B}_0 \leftarrow \{(\epsilon, 0)\}$
for $t = 1, \ldots, T$ **do**
  $\mathcal{C} \leftarrow \varnothing$
  for all $(z_{<t}, s) \in \mathcal{B}_{t-1}$ **do**
    $q_t(a) \leftarrow \log p_\theta(a \mid x, K, z_{<t})$ for ownership symbols a
    Compute $\mathcal{A}_t(z_{<t})$ by Equation 8
    for all $a \in \mathcal{A}_t(z_{<t})$ **do**
      Add $(z_{<t} \,\|\, a, s + q_t(a))$ to $\mathcal{C}$
    end for
  end for
  $\mathcal{B}_t \leftarrow \textsc{TopB}(\mathcal{C}, B)$
end for
$\widehat{z}_B \leftarrow \operatorname{argmax}_{(z,s) \in \mathcal{B}_T} s$
return $\widehat{z}_B$, $R(x, \widehat{z}_B)$

D Extended Experiments

D.1 Ownership-interface ablations and diagnostics

The adaptation-by-support comparison uses one common Qwen3.5-4B checkpoint, indexed EPOR prompt, canonical decimal codec, known K, evaluation set, and deterministic reconstruction. Adapted rows use the three task-adapted QLoRA adapters; frozen rows use no adapter or demonstrations. Constrained rows use the same completable-prefix mask, beam size four, and exactly T generated symbols as EPOR. Unmasked rows also request exactly T symbols, but the full vocabulary remains available. This removes the entire legality mask, not only the remaining-capacity condition.

Table 8: Search settings for the adaptation-by-support ablations. "LM beam" uses the decoder's original beam-score convention; no legal-subset renormalization is added.

Condition	Beam	Score normalization	Generated length	EOS	Codec validation
Adapted, constrained (EPOR)	4	LM beam score	exactly T	excluded as route action	atomic; reject unsupported
Frozen, constrained	4	LM beam score	exactly T	excluded as route action	atomic; reject unsupported
Adapted, unconstrained	4	LM beam score	exactly T	suppressed before T	retain as invalid (-1)
Frozen, unconstrained	4	LM beam score	exactly T	suppressed before T	retain as invalid (-1)

Unsupported symbols (including an unexpected EOS), noncanonical routes, missing sources, and wrong-length outputs are scored as emitted and are never repaired. Unit tests verify length preservation, illegal-symbol scoring, canonical-route checks, EOS handling, and deterministic evidence reconstruction. In the reported unmasked runs, every generated position nevertheless used an in-codec ownership symbol, so stable filtering consumed every input occurrence and evidence preservation remained 1.0. Their incomplete-structure scores instead come from routes that omit one or more of the known K sources. An unsupported symbol would be retained as invalid and would reduce evidence preservation if its position could not be assigned to an emitted source bucket.

Table 8 makes the matched and unmatched search support explicit for these four adaptation-by-support conditions.

For the fixed-template intervention, the causal LM sees the canonical sequence in implementation notation, $0,1,...,K-1,0,...$ at earlier ownership positions, truncated to the current prefix length. This sequence is independent of both the decoded and gold assignments. The decoder still computes its legal actions from the unmodified decoded route. The graded intervention preserves first appearances; once all K labels have appeared, it deterministically rotates a preselected fraction of model-visible prior labels. Thus both interventions preserve output support and reconstruction while selectively removing assignment-specific information from the causal context.

The matched fixed-template history ablation uses the same route-independent legal template during training and evaluation. Gold canonical ownership symbols remain the supervised targets, but the input side of the causal shift contains only the mixed-text prompt and the fixed template prefix. It therefore learns the same LM-token-level objective without observing example-specific $z_{<t}$. Backbone, QLoRA rank, learning rate, context length, examples, epochs, seeds, codec, beam budget, legality mask, and reconstruction are identical to EPOR. Evaluation again shows the fixed template to the LM while the external decoder maintains the true generated prefix solely to compute legal actions.

D.2 Full MultiNLI metrics

Table 9 supplements the main system comparison with position-level and boundary metrics.

Table 9: Full results on the controlled MultiNLI UnMixBench test set. MP-WER is mean±sample SD over three training runs; remaining entries are three-run means, with corresponding SDs provided in the supplementary results.

Method	MP-WER ↓	Ownership acc. ↑	Switch F1 ↑	Evidence ↑
Compact source-array generation	0.6696±0.0608	–	–	0.0845
Evidence-explicit source-array generation	0.6747±0.0242	–	–	0.0814
Turn-marker generation	0.4271±0.0096	0.6991	0.8081	0.3957
Global Transformer	0.5368±0.0192	0.6658	0.6847	1.0000
EMA prototype	0.5165±0.0140	0.6703	0.7390	1.0000
RGS-CRF	0.4799±0.0063	0.6917	0.7001	1.0000
Legal projection	0.4788±0.0140	0.6904	0.7695	1.0000
EPoR	0.3757±0.0056	0.7457	0.8514	1.0000

D.3 Generation Baselines: Prompt Variants and Error Diagnostics

Table 10: Generation-pipeline diagnostics, mean±sample SD over three runs. Conditional metrics use each method's successfully parsed subset.

Prompt	Parse	Limit hit	Conditional MP-WER	Conditional Evidence
Compact	0.6972±0.1147	n/a	0.5272±0.0108	0.1239±0.0259
Evidence-explicit	0.6852±0.0274	0.0034±0.0023	0.5256±0.0225	0.1189±0.0029

Table 10 separates parsing and truncation behavior from conditional reconstruction quality. All-example evaluation, including parse failures, is the reported comparison. The parser accepts only a nested JSON value with exactly K arrays of strings; invalid JSON or wrong cardinality is scored as an empty prediction, and no output is repaired. Termination metadata were not retained for the compact-prompt runs, so their limit-hit rate is unavailable rather than inferred. Conditional-on-parse metrics describe different method-specific subsets and are diagnostic rather than a replacement for the all-example comparison. Relative to evidence-explicit generation, EPoR changes all-example Evidence by +0.9186 (95% CI [0.8904, 0.9444]).

D.4 Task-matched turn-marker generation

Table 11 reports the three-run MultiNLI result. The first difference column is EPoR minus turn-marker generation; the second is turn-marker generation minus evidence-explicit source-array generation. Signed differences favor the row-named system when positive for ownership accuracy, Switch F1, and Evidence, but when negative for MP-WER. Confidence intervals use 20,000 source-group bootstrap draws after averaging training runs within each example.

Table 11: Task-matched turn-marker generation on the 2,640-example MultiNLI evaluation. The turn-marker column is mean±sample SD over three training runs. Difference columns give paired point estimates with 95% source-group bootstrap intervals.

Metric	Turn-marker generation	EPoR − Turn-marker	Turn-marker − source-array generation
MP-WER ↓	0.4271±0.0096	−0.0515 [−0.0636,−0.0395]	−0.2476 [−0.2720,−0.2238]
Ownership acc. ↑	0.6991±0.0068	+0.0466 [0.0381,0.0554]	–
Switch F1 ↑	0.8081±0.0032	+0.0432 [0.0338,0.0530]	–
Evidence ↑	0.3957±0.0131	+0.6043 [0.5617,0.6461]	+0.3143 [0.2778,0.3524]

The strict marker parser succeeds on 0.9831±0.0037 of examples, and only 0.0010±0.0011 reach the generation limit. Turn markers substantially improve on source-array generation by removing output reordering, yet lexical regeneration remains imperfect. On MultiNLI, EPoR retains advantages in reconstruction, occurrence-level ownership, boundary recovery, and exact evidence preservation. This system-level output-interface baseline isolates mixed-order marker generation from source-array reordering, but is not a representation-only comparison because it still regenerates every word.

Table 12 extends the comparison to all five tracks using the same three training seeds. Confidence intervals use 20,000 paired provenance-group bootstrap draws after averaging seeds within each example: source groups for MultiNLI, meetings for AMI and ICSI, and documents for ReadingBank. Digital resampling keeps the three schedules for each source group together; its mean and sample SD are computed from the equal-profile average within each seed. These pointwise intervals condition on the three trained checkpoints.

The paired MP-WER intervals favor EPoR on MultiNLI, AMI, ICSI, and ReadingBank. Digital has a small point difference in favor of turn-marker generation, with an interval spanning zero. Across all five tracks, EPoR has higher ownership accuracy and exact evidence preservation, whereas turn-marker generation sometimes changes the observed word multiset. Table 13 gives the ownership intervals and boundary scores after the lexical alignment in Appendix C.2. Boundary recovery favors turn-marker generation on AMI, ICSI, and Digital, and EPoR on MultiNLI and ReadingBank.

Table 12: **EPOR and turn-marker generation across five tracks.** Turn-marker MP-WER is mean±sample SD over three runs. Paired differences are EPOR minus turn-marker, with 95% group-bootstrap intervals; negative values favor EPOR. Ownership accuracy and Evidence columns give EPOR / turn-marker means.

Track	EPOR MP	Turn-marker MP	Paired difference [95% CI]	Ownership acc.	Evidence
MultiNLI	0.3757	0.4271±0.0096	-0.0515 $[-0.0636, -0.0395]$	0.7457 / 0.6991	1.0000 / 0.3957
AMI	0.6320	0.6928±0.0312	-0.0608 $[-0.0724, -0.0481]$	0.6205 / 0.4441	1.0000 / 0.6377
ICSI development	0.6161	0.6690±0.0185	-0.0529 $[-0.0704, -0.0353]$	0.6228 / 0.4805	1.0000 / 0.6600
ReadingBank	0.6156	0.6862±0.0154	-0.0706 $[-0.0859, -0.0556]$	0.6224 / 0.4959	1.0000 / 0.3325
Digital	0.6474	0.6449±0.0088	$+0.0026$ $[-0.0067, 0.0120]$	0.5912 / 0.5426	1.0000 / 0.7769

Table 13: **Ownership and boundary recovery after lexical alignment.** Differences are EPOR minus turn-marker, with paired 95% group-bootstrap intervals. Positive differences favor EPOR. All examples, including strict parsing failures, contribute to the reported scores.

Track	Ownership difference [95% CI]	Switch F1: EPOR / turn-marker	Switch F1 difference [95% CI]
MultiNLI	$+0.0466$ [0.0381, 0.0554]	0.8514 / 0.8081	$+0.0432$ [0.0338, 0.0530]
AMI	$+0.1765$ [0.1632, 0.1891]	0.2567 / 0.4469	-0.1902 [-0.2090, -0.1714]
ICSI development	$+0.1423$ [0.1258, 0.1572]	0.3018 / 0.5168	-0.2150 [-0.2379, -0.1885]
ReadingBank	$+0.1264$ [0.1121, 0.1412]	0.4459 / 0.4161	$+0.0298$ [0.0148, 0.0449]
Digital	$+0.0486$ [0.0406, 0.0568]	0.6764 / 0.7136	-0.0372 [-0.0500, -0.0246]

Repeated words can admit multiple optimal lexical alignments. Reversing the fixed tie preference changes each track's mean ownership accuracy by less than 0.0006 and mean switch F1 by less than 0.0003. The comparisons are unchanged under this alternative convention.

D.5 Frontier zero-shot language models

Table 14 records the executed provider configuration for every zero-shot row in Table 2. The requested model alias, provider route, thinking setting, temperature, and output budget are reported for each scope. All calls used the same evidence-explicit prompt and parser.

The two Table 2 rows marked "mixed APIs" combine a locked MultiNLI DeepSeek-API control with later Alibaba Cloud transfer calls. They therefore report the executed cells under the same requested model name, while retaining their provider and budget differences in the Table 14. DeepSeek-V4.1-Flash was requested as `deepseek-flash`.

The four evaluations in Table 15 use the same strict evidence-explicit prompt, controlled-MultiNLI examples, parser, non-thinking mode, temperature, and generous output budget. Unresolved API errors are zero after retries; format failures remain measured failures. The DeepSeek-V4-Pro entry is the later single-provider full-budget rerun and is distinct from the mixed-API V4-Pro row in Table 2. Ownership accuracy is undefined for free-text source arrays. Transfer-view summaries and hashes are provided in the supplementary experiment package.

D.6 Ownership adaptation and history

The adapted conditions in Table 3 average three models; the frozen conditions use the same unadapted checkpoint. All conditions use the same batch-one evaluation setting as the MultiNLI EPOR row.

The matched route-independent fixed-template history ablation averages three runs and differs from EPOR only in the ownership prefix visible to the causal LM during training and evaluation. The fixed-template-minus-EPOR differences are +0.0730 MP-WER (95% CI [0.0620, 0.0840]), -0.0345 ownership accuracy ([-0.0419, -0.0271]), and -0.0911 switch F1 ([-0.0982, -0.0842]). Evidence preservation and complete K-way support remain 1.0 in every run. This matched training-and-evaluation comparison tests assignment-specific causal history while holding the decoder and reconstruction fixed.

Table 14: Hosted-model execution configuration for Table 2. "Scope" identifies the cells supplied by each run: MNLI, AMI, ICSI-dev, ReadingBank (RB), or the three-profile digital aggregate (DIG).

Table row / scope	Provider	Requested model ID	Thinking	Temp.	Completion-token budget
Qwen3.8-Flash / all	Alibaba Cloud Model Studio	`qwen3.8-flash`	off	0	$\max(8192,64T)$
Qwen3.8-Max / all	Alibaba Cloud Model Studio	`qwen3.8-max`	off	0	$\max(8192,64T)$
DeepSeek-V4-Flash / MNLI	DeepSeek API	`deepseek-v4-flash`	off	0	$\max(256,16T)$
DeepSeek-V4-Flash / AMI, ICSI-dev, RB, DIG	Alibaba Cloud Model Studio	`deepseek-v4-flash`	off	0	$\max(8192,64T)$
DeepSeek-V4-Pro / MNLI	DeepSeek API	`deepseek-v4-pro`	off	0	$\max(256,16T)$
DeepSeek-V4-Pro / AMI, ICSI-dev, RB, DIG	Alibaba Cloud Model Studio	`deepseek-v4-pro`	off	0	$\max(8192,64T)$
DeepSeek-V4.1-Flash / AMI, ICSI-dev, DIG	DeepSeek API	`deepseek-flash`	off	0	$\max(8192,64T)$
DeepSeek-V4.1-Flash / MNLI, RB	DeepSeek API	`deepseek-flash`	off	0	$\max(8192,64T)$

Table 15: MultiNLI compliance diagnostics for four hosted evaluations under the generous $\max(8192,64T)$ budget. Metrics include all examples. Parse failures and provider refusals receive no repair.

System	Temp.	JSON parse ↑	Limit hit ↓	Refusal ↓
DeepSeek-V4.1-Flash	0	0.9284	0.0000	0.0000
DeepSeek-V4-Pro	0	0.8811	0.0000	0.0000
Qwen3.8-Flash	0	0.8023	0.0000	0.0004
Qwen3.8-Max	0	0.9826	0.0000	0.0000

Table 16: **Inference-time ownership-history interventions on MultiNLI.** The checkpoint is fixed; only the prefix shown to the scorer changes. The controller uses the actual candidate prefix, leaving output legality intact.

Condition	MP-WER ↓	Ownership acc. ↑	Switch F1 ↑	Evidence ↑	Complete K-way ↑
Unmodified history	0.3729	0.7457	0.8514	1.0000	1.0000
Route-independent template	0.9126	0.4722	0.5057	1.0000	1.0000
25% eligible-label rotation	0.4761	0.6702	0.7763	1.0000	1.0000
50% eligible-label rotation	0.5335	0.6263	0.7080	1.0000	1.0000
100% eligible-label rotation	0.5941	0.5796	0.5692	1.0000	1.0000

Table 17: Matched representation comparison. Recovery values are three-run means; both models have 64,929,792 trainable parameters.

Representation	MP-WER ↓	LM train sequences	Rel. train time ↓
Canonical (EPoR)	0.3757	60,000	**1.00**×
Arbitrary labels (PI-Route)	0.3782	602,608	4.08×

Table 16 reports the numerical outcomes of the inference-time interventions, while Figure 4 shows that recovery degrades as more assignment-specific history is corrupted. For the model used in this curve, MP-WER rises from 0.3729 without corruption to 0.5941 under full eligible-label rotation. Evidence and complete K-way support remain exact, so the change is semantic rather than a legality failure. The curve uses one trained checkpoint and an inference-time intervention, which demonstrates sensitivity, not a uniquely identified memory mechanism.

D.7 Permutation-invariant representation comparison

Table 18: Full matched representation comparison on MultiNLI, mean±sample SD over three training runs.

Method	MP-WER ↓	Ownership acc. ↑	Switch F1 ↑	Evidence ↑	Structure ↑
EPoR	0.3757±0.0056	0.7457±0.0039	0.8514±0.0022	1.0000	1.0000
PI-Route	0.3782±0.0012	0.7438±0.0037	0.8521±0.0023	1.0000	1.0000

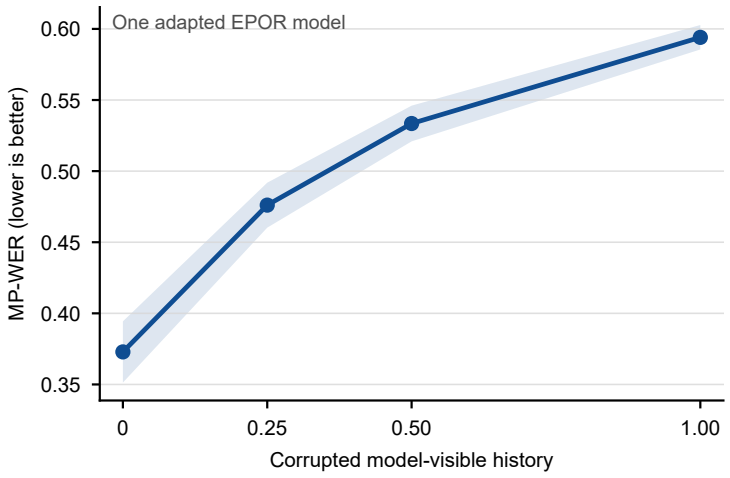


Figure 4: **Sensitivity to model-visible ownership history.** Points report one adapted model; bands are 95% source-group bootstrap intervals over 240 groups, retaining all eleven interleavings within each group. Corruption begins only after all K labels have appeared; the decoder's true legal prefix is unchanged.

Tables 17 and 18 report the compact cost comparison and the full recovery metrics, respectively. All paired recovery intervals span zero. PI-Route has the same 64,929,792 trainable parameters but requires 4.08× the mean training wall time. The measured 602,608 versus 60,000 LM sequence evaluations (10.04×) count the implemented two-pass training procedure: all permuted routes are scored without gradients, then the chunks containing the minimum-loss routes are replayed for backpropagation. This counter therefore includes gradient-replay rows and is distinct from the single-pass combinatorial average $\sum_i K_i!/N$. All six training runs used the same NVIDIA RTX 5090 host and two-epoch, 10,000-example training loop. Mean recorded training time was 7,005.13 seconds for EPOR and 28,565.23 seconds for PI-Route; thus EPOR used 75.5% less measured wall time in this implementation. The timing covers model training (including permutation scoring and gradient replay for PI-Route), not evaluation, model acquisition, or data preparation.

D.8 Difficulty factors

D.8.1 Mixing intensity

Figure 3 shows that recovery difficulty is not monotonic in mixing intensity. For $(K,T) = (2,16)$, MP-WER rises from 0.053 at $M = 0$ to 0.249 at $M = 0.4$, then falls to 0.094 at $M = 1$. With two sources, switching at every position restricts the route to an alternating pattern, so maximal switching can reduce structural uncertainty. The larger joint conditions remain substantially harder: at $M = 1$, MP-WER is 0.448 for $(3,24)$ and 0.634 for $(4,32)$. Because K and T vary together, cross-curve differences describe joint task conditions rather than an isolated source-count effect.

D.8.2 Source similarity

Table 19 compares the four frozen-encoder source-similarity bands after exact matching on K, T, and target M. Each (K,T,M) cell receives equal weight. Intervals use 20,000 grouped bootstrap draws over the 240 source groups while retaining all mixing views and averaging the three trained models within each example.

Table 19: **Recovery across matched source-similarity strata.** Entries are matched means with 95% source-group bootstrap intervals. Lower MP-WER is better; higher values are better otherwise.

Similarity band	MP-WER ↓	Ownership acc. ↑	Switch F1 ↑
Low	0.3419 [0.3227, 0.3619]	0.7747 [0.7610, 0.7882]	0.8657 [0.8515, 0.8790]
Medium	0.3775 [0.3513, 0.4035]	0.7494 [0.7317, 0.7673]	0.8451 [0.8259, 0.8641]
High	0.3888 [0.3665, 0.4115]	0.7323 [0.7151, 0.7491]	0.8487 [0.8358, 0.8613]
Extreme	0.3944 [0.3730, 0.4163]	0.7266 [0.7100, 0.7428]	0.8461 [0.8298, 0.8616]

The extreme-minus-low contrasts are +0.0524 MP-WER (95% CI [0.0230, 0.0818]), -0.0481 ownership accuracy ([-0.0693, -0.0268]), and -0.0196 switch F1 ([-0.0406, 0.0013]). The switch-F1 interval spans zero; the other two contrasts resolve in the direction of harder recovery for the extreme-similarity stratum. The independent encoder cosine is an operational stratification measure rather than a universal semantic scale, and this descriptive analysis does not identify a causal similarity effect.

D.9 Construction and backbone details

Table 20 verifies that the clock generator induces different route statistics while holding the source words, K, and T fixed. Both renderings have mean $T = 24.0$, balanced K, zero source- length imbalance, mean lexical overlap 0.0251, and mean $\log_2$ route-space 36.6228.

Table 20: Route statistics for paired controlled and independent-clock interleavings.

Construction	Mixing M	Switches	Mean run	Run var.	Ambiguity
Controlled	0.4953	12.4542	2.6454	0.4096	0.1208
Independent clock	0.5651	14.4542	1.7185	1.0254	0.1583

Table 21: Full cross-backbone comparison on 240 paired source groups. Each backbone uses one training run. Evidence reports controlled/clock; route methods have structural validity 1.0 in both conditions.

Backbone	Output	Controlled MP-WER ↓	Clock MP-WER ↓	Evidence C/I ↑
Qwen3.5-4B	EPoR	0.4163	0.5189	1.0000/1.0000
	RGS-CRF	0.5344	0.6075	1.0000/1.0000
	Free generation	0.6012	0.6480	0.1708/0.1792
Qwen3.5-9B	EPoR	0.3625	0.4459	1.0000/1.0000
	Free generation	0.5368	0.5886	0.2125/0.2333
Gemma-7B-IT	EPoR	0.4630	0.5488	1.0000/1.0000
	Free generation	0.6449	0.6832	0.0583/0.0417

Table 21 shows the ownership-over-generation ordering for all three evaluated backbones. For Qwen3.5-4B, EPoR minus free generation is -0.1292 MP-WER on independent clocks (95% CI [-0.1697,-0.0895]). The Gemma-7B-IT difference is -0.1344 ([-0.1695,-0.0998]), and the Qwen3.5-9B difference is -0.1427 ([-0.1801,-0.1058]). Within EPoR, 9B minus 4B is -0.0538 on controlled data ([-0.0789,-0.0292]) and -0.0730 on clocks ([-0.1000,-0.0467]). The 9B model yields lower EPoR MP-WER on both constructions.

Table 22: Unequal-length construction comparison on 180 paired source groups. †The final row receives gold source lengths and is a privileged reference rather than a deployable baseline.

Method	Clock MP-WER	Long-run MP-WER	Bursty MP-WER	Min. evidence ↑
EPoR	0.500	0.288	0.469	1.000
RGS-CRF	0.568	0.347	0.612	1.000
Task-finetuned source-array generation	0.623	0.658	0.592	0.139
Random legal route	0.726	0.775	0.697	1.000
Equal contiguous blocks	0.683	0.278	0.759	1.000
Gold-length contiguous blocks†	0.668	0.000	0.740	1.000

Table 24: Source-array and structured baselines by serialization view. Entries are mean MP-WER over three training seeds. "Free-c" and "Free-e" denote compact and evidence-explicit task-finetuned generation. Δ compares EPOR with the lowest-mean source-array or structured baseline shown in that view; 95% CIs resample provenance groups after averaging seeds. Negative differences favor EPOR. Best and second-best means are bold and underlined.

View	Free-c	Free-e	Global	EMA	RGS-CRF	Projection	EPOR	Δ to best alternative [95% CI]
AMI	0.7275	0.7418	<u>0.6406</u>	0.6596	0.6483	0.6431	**0.6320**	$-0.0086\,[-0.0175, +0.0003]$
ICSI development	0.7439	0.7530	<u>0.6389</u>	0.6674	0.6495	0.6483	**0.6161**	$-0.0228\,[-0.0377, -0.0078]$
ReadingBank	0.8415	0.8547	0.6820	0.7427	<u>0.6555</u>	0.6843	**0.6156**	$-0.0399\,[-0.0517, -0.0281]$
Digital–smooth	0.7193	0.7283	0.7718	0.7326	0.7467	**0.6741**	<u>0.6750</u>	$+0.0009\,[-0.0128, +0.0152]$
Digital–bursty	0.7431	0.7307	0.6763	0.6794	<u>0.6406</u>	0.6764	**0.6156**	$-0.0251\,[-0.0409, -0.0090]$
Digital–unequal-rate	0.7306	0.7352	0.7482	0.7114	0.7223	<u>0.6788</u>	**0.6517**	$-0.0270\,[-0.0413, -0.0130]$

Table 22 shows that EPOR has the lowest MP-WER among the task-trained systems under all three unequal-length constructions. The equal-block heuristic is slightly better on the long-run construction (0.278 versus 0.288), while its substantially larger error on clock and bursty routes exposes the construction-specific shortcut.

D.10 Domain transfer

Without target-domain tuning, EPOR lowers MP-WER relative to RGS-CRF on WikiText, the shared AMI view, and the ICSI reserve but increases it on AG News, indicating non-uniform domain transfer.

Table 23: Domain transfer, mean$\pm$SD over three training runs. WikiText and AG News intervals resample examples; AMI and ICSI-reserve intervals resample meetings. AMI is the same frozen 1,000-window view used in the temporal-serialization analysis; ICSI here denotes the separate 300-window consumed reserve. Δ is EPOR minus RGS-CRF.

Domain	RGS-CRF MP-WER	EPOR MP-WER	Δ MP-WER [95% CI]
WikiText	0.3583$\pm$0.0081	0.2506$\pm$0.0136	-0.1077 [-0.1174, -0.0982]
AG News	0.6187$\pm$0.0220	0.7322$\pm$0.0228	0.1135 [0.1056, 0.1213]
AMI	0.6483$\pm$0.0103	0.6320$\pm$0.0086	-0.0163 [-0.0257, -0.0062]
ICSI reserve	0.6581$\pm$0.0036	0.6283$\pm$0.0057	-0.0297 [-0.0457, -0.0113]

D.11 Serialization transfer

Table 24 decomposes the source-array and structured-baseline serialization aggregates in Table 2; the turn-marker comparison is in Table 12. The digital aggregate is the equal-profile mean over smooth, bursty, and unequal-rate schedules applied to the same 400 source groups. The ICSI row is the 400-window development view, not the 300-window reserve in Table 23.

Among the systems in Table 24, EPOR has the lowest mean MP-WER on five of the six views. On the smooth digital profile, legal projection is lower by 0.0009 and the paired interval spans zero. Against the per-view runner-up, the intervals exclude zero for ICSI development, ReadingBank, bursty digital, and unequal-rate digital, but not for AMI. Hosted outcomes remain in Table 2, with compliance diagnostics in Table 15.

D.12 Candidate quality and ranking

Figure 5 separates candidate generation from candidate ranking. At beam 32, the best candidate under gold MP-WER reaches 0.2031, whereas the model score selects 0.3737, a regret of 0.1706. The search can therefore retain candidates with substantially lower reconstruction error than the route selected by the learned score. This gap is measured within the finite candidate set; it is not a bound on the error of exact score maximization. The oracle uses the reference sources and is not deployable. All candidates satisfy the same structural constraints, regardless of beam width. A reference-free reranker over complete (x,z) candidates is a possible extension; the reported selector is unchanged.

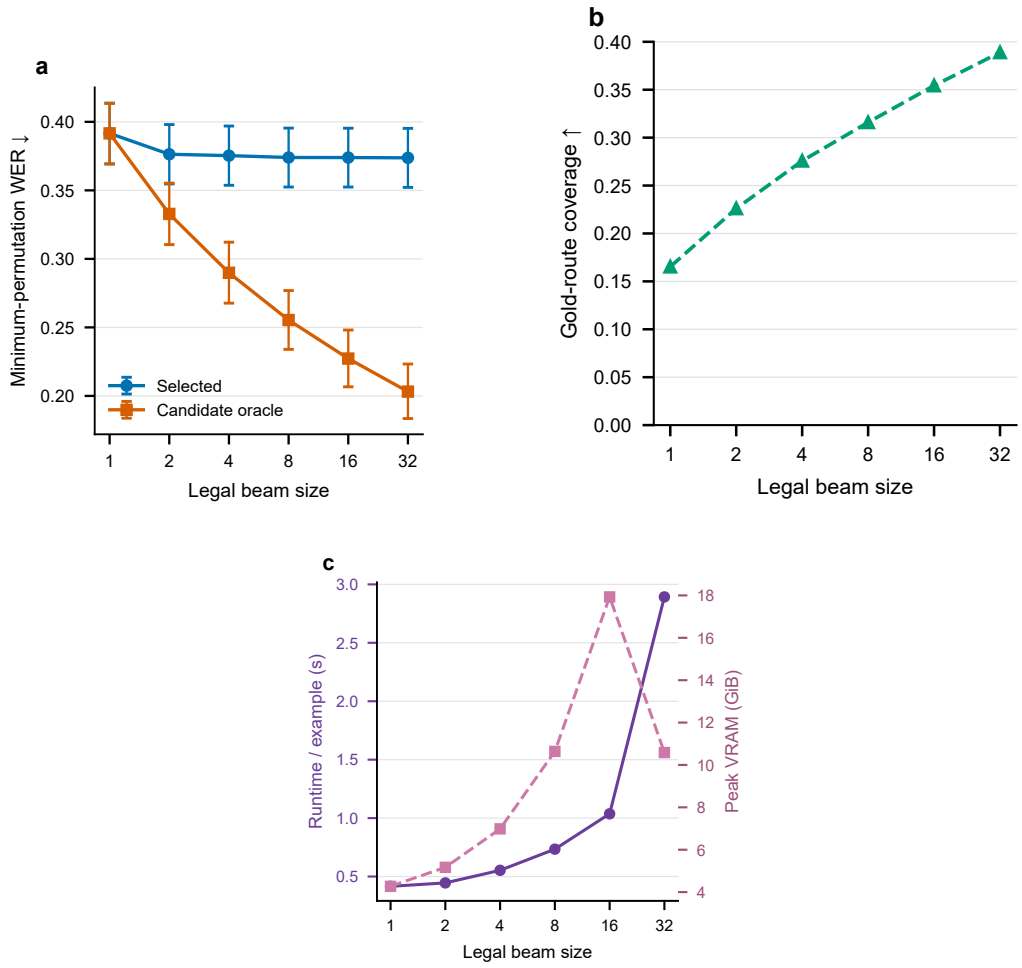


Figure 5: **Candidate quality, selection, and cost.** Selected and candidate-oracle MP-WER are shown across legal beam sizes, together with gold-route coverage, runtime, and peak allocated memory. Points average three trained models; intervals resample source groups.

E Extended Related Work

Word-level text unmixing is also related to several established problem families outside the closest NLP settings reviewed in Section 2. These connections concern different aspects of the problem: formal-language work studies order-preserving interleavings of discrete sequences, document understanding studies recovery of reading structure from spatial and visual evidence, and blind source separation studies recovery of latent sources from observed mixtures. They help situate text unmixing among broader source-recovery and de-interleaving problems.

E.1 Shuffle Languages and String Unshuffling

Order-preserving sequence interleaving has a classical formulation in formal-language theory through the *shuffle product*. For strings u and v, their shuffle contains the strings obtainable by interleaving symbols from u and v while preserving the internal order of each string. Early work studied the corresponding *merge recognition* problem: given an observed string and candidate component strings, determine whether the former is an order-preserving merge of the latter (Mansfield, 1982). For any fixed number of given component strings, merge recognition admits a polynomial-time algorithm, whereas allowing the number of components to vary as part of the input makes the problem NP-complete (Mansfield, 1983). The shuffle product has subsequently been studied more broadly as an

operation in formal-language theory and combinatorics on words, including its expressive properties and connections to models of concurrent processes (Berstel et al., 2010; Restivo, 2015).

A related line of work considers *shuffle squares*, where the component string itself is unknown but the two components are constrained to be identical. Given a string w, the problem asks whether there exists some u such that $w \in u \sqcup\!\sqcup u$. Recognizing such shuffle squares is NP-complete (Rizzi & Vialette, 2013; Buss & Soltys, 2014).

These problems share with word-level text unmixing the combinatorial structure of order-preserving interleaving, but differ in what is observed and what must be inferred. Classical merge recognition is supplied with the candidate component strings, while shuffle-square recognition replaces this information with the strong constraint that the two latent components are identical. In word-level text unmixing, by contrast, all K source sequences are latent, need not be identical, and have exchangeable source identities. The observation preserves the individual word occurrences but not their source ownership, so recovery requires inferring an occurrence-level partition from the lexical stream. The classical complexity results above therefore do not directly establish the computational complexity of word-level text unmixing; rather, shuffle theory provides a formal analogue of its underlying order-preserving interleaving structure.

E.2 Reading-Order Recovery in Document Understanding

Document analysis provides another closely related setting in which serialization can obscure latent textual structure. Reading-order detection has long been treated as part of recovering the logical organization of complex documents. Earlier work developed representations and evaluation procedures for reading order over document regions, including layouts with ordered, unordered, and grouped relations (Clausner et al., 2013). More recent document-understanding systems learn reading structure using textual features together with explicit spatial or visual information. LayoutReader predicts reading order from text and layout representations (Wang et al., 2021), while ERNIE-Layout incorporates reading-order prediction into multimodal pretraining over textual, spatial, and visual features (Peng et al., 2022). Other approaches directly address incorrect document serialization through token-path prediction (Zhang et al., 2023) or model reading order as ordering relations among layout elements rather than as a single global permutation (Zhang et al., 2024).

The spatial track of UNMIXBENCH is motivated by the same broad failure mode: multiple coherent reading flows can be serialized into a single stream in an order that does not preserve those flows. The information interface is different, however. For the ReadingBank track, document metadata are used during benchmark construction to derive deterministic operational reading flows and the corresponding ownership routes, whereas the evaluated model receives only the resulting lexical stream x and source count K. Layout coordinates, page geometry, visual features, and the reading-order metadata used for construction are not supplied to the model.

The spatial text-unmixing track should therefore not be interpreted as an alternative to layout-aware reading-order prediction. Reading-order systems aim to exploit surviving document geometry and multimodal structure to determine how page elements should be traversed. Text unmixing instead studies a downstream recovery setting in which such attribution information has already been discarded while the lexical occurrences themselves remain available. It asks whether the latent reading flows can still be recovered from the serialized lexical evidence alone.

E.3 Conceptual Relation to Blind Source Separation

A broader conceptual connection arises with blind source separation (BSS), which studies recovery of latent source signals from observed mixtures without direct access to the individual sources. Early adaptive methods sought to separate unknown independent sources from sensor observations (Jutten & Hérault, 1991). Independent component analysis subsequently provided a statistical formulation based on recovering components with reduced statistical dependence (Comon, 1994), while information-theoretic approaches demonstrated blind separation by maximizing information transfer (Bell & Sejnowski, 1995). Classical surveys and algorithmic treatments further developed the statistical principles and practical methods underlying BSS and independent component analysis (Cardoso, 1998; Hyvärinen & Oja, 2000).

The relation to word-level text unmixing is conceptual rather than algebraic. Classical BSS generally observes superpositions or transformations of simultaneously active source signals and recovers the underlying signals under assumptions about the sources and mixing process, often including statistical independence. Word-level text unmixing does not assume an additive mixture, a linear mixing matrix, or an ICA-style independence model. Instead, the mixing mechanism is serialization: discrete word occurrences from several sources are interleaved into a shared sequence while each occurrence itself remains intact. What is lost is the correspondence between an occurrence and its latent source.

Thus, text unmixing is better viewed as a discrete source-attribution problem under order-preserving interleaving than as an instance of classical BSS. The broader connection is that both settings separate evidence produced by latent sources after source identity has become entangled in the observation process. In the text-unmixing setting studied here, this abstraction is specialized to complete lexical evidence with missing ownership.