DATA-EFFICIENT NARRATIVE KNOWLEDGE INJECTION VIA ENTITY-LEVEL DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models often struggle to acquire knowledge directly from narrative corpora, while existing approaches rely on extensive synthetic augmentation to make the injected knowledge accessible. We attribute this difficulty to a binding–access mismatch: autoregressive training binds knowledge to highly specific, left-to-right contexts, whereas inference-time queries may provide substantially different cues. We introduce EntDiff, a data-efficient approach that learns primarily from source narratives with only lightweight preprocessing. Compared with augmentation-based methods, EntDiff generates more than three orders of magnitude fewer tokens during data preparation and trains on 2.5–6$\times$ fewer tokens. EntDiff reshapes how narrative knowledge is bound during training by making entity references explicit, limiting dependence on overly specific contexts, and creating diverse bidirectional associations accessible from different inference-time cues. Experiments show that EntDiff matches or outperforms strong augmentation-based baselines on dataset-paired question answering while substantially improving episodic recall. Our code is available in the anonymous repository at `https://anonymous.4open.science/r/EntDiff-E4BC`.

1 INTRODUCTION

A person who lives through a first-person experience encodes it as *episodic memory*, a record of what happened in what situation (Tulving et al., 1972). Later, the memory can be reconstructed from a *cue*, a piece of information linked to the original experience such as a place, a song, or a smell (Tulving, 1983). This ability to encode and recall an experience is a defining property of human memory. A mirroring task arises naturally for language models: after memorizing a long narrative story, the model is asked to recall a specific event given a cue.

Background: Previous studies have found that standard fine-tuning on the original corpus struggles to make newly learned knowledge reliably extractable (Allen-Zhu & Li, 2024), and that knowledge becomes reliably accessible only when the original corpus is presented in many augmented variants during training (Ovadia et al., 2024). Following this idea, subsequent approaches construct much larger synthetic training corpora from the source content using rephrasing, question–answer generation, entity-relation synthesis, and other augmentation strategies (Ovadia et al., 2024; Yang et al., 2025; Lin et al., 2026; Tang et al., 2026). For example, EntiGraph (Yang et al., 2025) decomposes stories into "entity-relation graph" descriptions, expanding 1.5M source tokens into 455M synthetic tokens. In contrast, humans can remember what happens in a narrative story after reading the source narrative alone, suggesting that the substantial cost of large-scale synthetic augmentation may partly compensate for the data inefficiency of the current knowledge-injection paradigm.

Beyond data inefficiency, many augmentation-based approaches inject knowledge in a patch-style manner by transforming the original narrative into localized semantic facts. For example, SynthQA (Lewis et al., 2021; Mecklenburg et al., 2024) converts articles into standalone question–answer pairs, while EntiGraph (Yang et al., 2025) converts narrative content into isolated entity-entity relations. Such representations make individual pieces of knowledge easier to retrieve from likely downstream queries and can therefore perform well on dataset-paired QA. However, in doing so, they may wash out the situational context, fine-grained details, and narrative continuity of the original story. These elements are essential for episodic knowledge, the temporally and situationally

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.

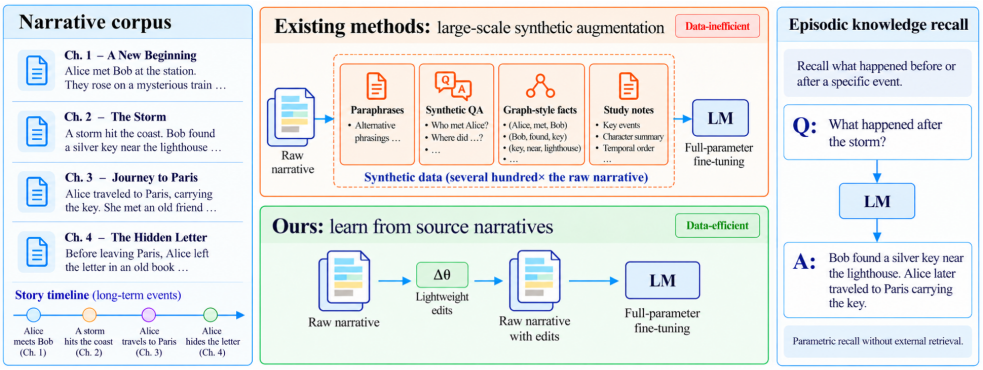


Figure 1: Existing knowledge-injection methods rely on large-scale synthetic augmentation, whereas EntDiff learns primarily from lightly preprocessed source narratives.

grounded knowledge of events and their surrounding context, including what happened before or after a particular event (Tulving et al., 1972). Consequently, a model may answer semantic questions such as "Who is X?" while failing to recall "What happened before or after X left the house that night?"

Moreover, such patch-style knowledge injection is highly coverage-dependent: knowledge is reliably accessible only when the augmented training data sufficiently cover the relevant query patterns. In contrast, learning primarily from the source narrative offers a more general approach, allowing the acquired knowledge to be accessed from diverse queries without exhaustive augmentation. Figure 1 summarizes this contrast between large-scale synthetic augmentation and our source-narrative-based learning paradigm, together with the episodic recall setting considered in this work.

Our scope and method: In this paper, we explore a distinct paradigm for knowledge injection: we fine-tune the LLM primarily on source narrative content with only lightweight preprocessing, enabling it to capture episodic knowledge while avoiding the substantial cost of large-scale synthetic augmentation. To achieve this, we revisit a fundamental question:

> *Why does autoregressive fine-tuning on the original corpus fail to make knowledge reliably accessible?*

We attribute this failure to a binding–access mismatch between training and inference: autoregressive training learns knowledge under specific left-to-right contexts, whereas inference-time queries may provide substantially different cues. This mismatch can make learned knowledge difficult to access, especially in long narratives.

EntDiff is designed specifically to mitigate this binding–access mismatch. Each component addresses a different source of the mismatch. Entity processing resolves ambiguous pronouns to explicit entity names and identifies named-entity spans, providing more stable targets for knowledge binding. Segmentation restricts each binding to a compact local context, reducing dependence on long and overly specific training cues. Entity-level diffusion removes the left-to-right constraint of autoregressive training, creating diverse cue-to-knowledge associations from bidirectional context under different corruption patterns. Jointly masking tokens within the same named-entity span additionally prevents trivial within-entity associations. Together, these components make knowledge learned from the source narrative accessible from a broader range of inference-time queries.

Contributions: Our main contributions are threefold.

- We identify a binding–access mismatch in autoregressive knowledge injection: knowledge learned under specific, left-to-right training contexts can be difficult to access from different inference-time cues.
- We propose EntDiff, a data-efficient approach that learns primarily from source narratives with lightweight preprocessing by reshaping how knowledge is bound during training, promoting entity-consistent bindings and more flexible cue-to-knowledge associations.
- Experiments show that EntDiff matches or outperforms augmentation-based methods on dataset-paired QA and substantially improves episodic recall, while generating more than

three orders of magnitude fewer tokens during data preparation and training on 2.5–6× fewer tokens.

2 Related Work

Knowledge Injection from Small Corpora. A long line of work studies how to inject the knowledge of a target corpus into the parameters of an LLM, so that it can be queried closed-book. However, directly training on the raw text struggles to make new knowledge reliably accessible (Ovadia et al., 2024; Allen-Zhu & Li, 2024). To overcome this problem, the dominant paradigm is *synthetic augmentation*: the source content is transformed into a much larger training corpus containing diverse derived views of the same knowledge. Common strategies include summaries (Ovadia et al., 2024), fact-based question–answer pairs (Mecklenburg et al., 2024), outputs of fixed prompt templates (Tang et al., 2026), and model-generated study notes (Lin et al., 2026). Beyond prompt-based augmentation, SEAL (Zweiger et al., 2025) uses reinforcement learning to optimize synthetic generation for downstream performance; LaPael (Kang et al., 2024) paraphrases in latent space by perturbing early layers; and EntiGraph (Yang et al., 2025) synthesizes text over an entity graph, expanding a 1.5M-token corpus into 455M synthetic tokens. Our work departs from this synthetic-augmentation paradigm by training primarily on source narrative content with lightweight preprocessing rather than on a massively expanded synthetic corpus.

Diffusion Language Models. Diffusion language models (dLLMs) replace the left-to-right prediction of autoregressive (AR) models with a masked denoising objective: a random subset of tokens is masked and recovered from the remaining tokens on both sides. Because the masked positions are randomly selected, each token is predicted from many varied contexts rather than one fixed prefix (Li et al., 2022; Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024). This any-order association naturally alleviates the *reversal curse*, where an AR model trained on "A is B" fails on the reversed query "B is A" (Berglund et al., 2024): dLLMs answer such reversed queries far better than strong AR models and improve reasoning and planning that left-to-right factorization struggles with (Nie et al., 2025; Ye et al., 2025a). The paradigm now scales to modern LLM sizes, whether trained from scratch (LLaDA, 8B, rivaling LLaMA3 8B (Nie et al., 2025)) or adapted from pretrained AR models (DiffuLLaMA from LLaMA (Gong et al., 2025); Dream 7B from Qwen2.5-7B (Ye et al., 2025b; Team, 2024)). Prabhudesai et al. (2025) demonstrate that in a data-constrained setting, dLLMs outperform AR on general tasks such as commonsense reasoning and reading comprehension. This any-order association is reminiscent of cue-based reconstruction in human episodic memory, where a fragment can serve as a retrieval cue for the surrounding event (Tulving, 1983). We exploit this characteristic to inject episodic knowledge from long narratives.

3 Method

Given a corpus of N long narratives $\mathcal{D} = \{D_j\}_{j=1}^{N}$, where each narrative D_j is a sequence of tokens, our goal is to adapt the parameters of a language model so that it can answer questions about the narratives in a closed-book setting, without retrieving or conditioning on the source documents at inference time.

3.1 Motivation: Binding–Access Mismatch

Prior work (Ovadia et al., 2024; Allen-Zhu & Li, 2024) has shown that autoregressive training often fails to make newly injected knowledge reliably accessible at inference time. Consider a narrative $D = (d_1, \ldots, d_T)$, where we drop the narrative index j for brevity. As illustrated in Figure 2, for a target token d_t, autoregressive training uses its preceding context $d_{<t} = (d_1, \ldots, d_{t-1})$ as the cue. Attention aggregates this cue into a hidden representation $\mathbf{h}_t = f_{\text{attn}}(d_{<t})$, and the model learns to predict d_t from this context-specific representation. We use *binding* to refer to this learned association between a piece of knowledge and the contextual representation from which it is predicted during training. At inference time, however, a query provides a different cue and produces a different representation $\mathbf{h}_q$, making d_t difficult to access.

We identify two sources of this mismatch. First, the training-time cue can be overly complex. In a long narrative, the cue for d_t may consist of a long prefix $d_{<t}$, whereas an inference-time query

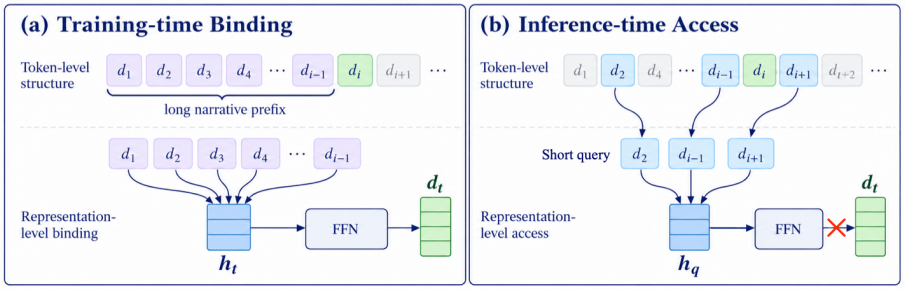


Figure 2: Illustration of the binding–access mismatch in autoregressive training. During training, the preceding context $d_{<t}$ serves as the cue for the target token d_t. Attention aggregates this cue into a hidden representation $\mathbf{h}_t$, from which the model learns to predict d_t. During inference, the query provides a different cue and produces a different hidden representation $\mathbf{h}_q$. When the inference-time cue differs substantially from the training-time cue, d_t cannot be reliably accessed.

typically contains only a small number of tokens. The two cues can therefore induce substantially different hidden representations, making the binding difficult to access.

Second, autoregressive training imposes a strong positional constraint on the cue used for binding. For each target token d_t, its cue is restricted to the prefix $d_{<t}$ and therefore contains only information appearing before d_t. In contrast, an inference-time cue may be formed from any content relevant to d_t, regardless of where that content appeared in the original narrative. This mismatch between training-time binding cues and inference-time access cues makes narrative knowledge difficult to retrieve reliably.

3.2 Proposed Approach

Figure 3 provides an overview of EntDiff, which consists of entity processing, segmentation, and entity-level diffusion. The three components progressively make entity references explicit, restrict the context used for knowledge binding, and create diverse bidirectional cue-to-knowledge associations.

3.2.1 Entity Processing

Narrative knowledge is often expressed through named entities and their interactions. Ambiguous pronouns can make it unclear which entity an event or description refers to, while entity-level diffusion requires explicit named-entity spans for joint masking. We therefore preprocess each narrative in two lightweight steps: pronoun resolution makes ambiguous references explicit, and named entity identification locates the named entities whose token spans are used during diffusion training.

Pronoun Resolution. We first locate candidate pronouns using deterministic regex-based rules and assign each candidate an index. We then provide the indexed candidates together with the narrative context to Qwen2.5-7B-Instruct. For each candidate, the model returns either the corresponding entity name or `SKIP`; it does not generate a rewritten version of the narrative. We replace only the pronouns that are resolved to explicit entity names, leaving the remaining text unchanged. For example, given "Ann Clotilde warned Jonathan not to approach the village. Jonathan ignored the warning and continued toward the coast. After several other events, she eventually decided to follow him," we replace "she" with "Ann Clotilde" and "him" with "Jonathan."

Named Entity Identification. After pronoun resolution, we first extract candidate entity strings with a deterministic capitalization-based extractor and then use Qwen2.5-7B-Instruct to categorize each candidate. We retain candidates assigned to `PERSON`, `PLACE`, `ORG`, or `BOOK_ENTITY`, and map their mentions to token spans in the processed narrative. These named-entity spans are used only to define joint entity masks during diffusion training and do not otherwise modify the source text.

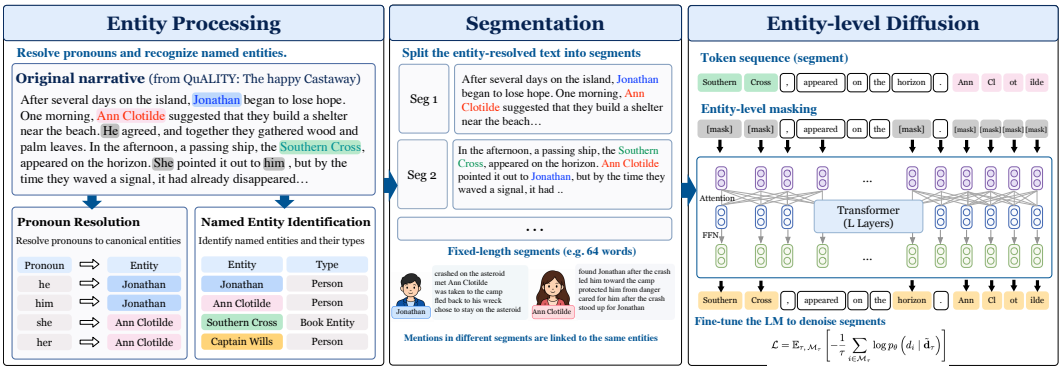


Figure 3: Overview of EntDiff. Entity Processing resolves ambiguous pronouns to explicit entity names and identifies named-entity spans, providing stable targets for subsequent knowledge binding. Segmentation restricts the context to which knowledge is bound, making the knowledge easier to retrieve from relevant cues. Entity-level diffusion creates diverse cue-to-knowledge bindings from bidirectional context under different corruption patterns, while jointly masking tokens of the same entity to avoid unnecessary within-entity token associations.

3.2.2 Segmentation

Training on the entire narrative allows knowledge to be bound to a large amount of surrounding context, including distant and irrelevant information. Such broad bindings make it difficult to retrieve the knowledge from a small set of relevant cues at inference time. We therefore use segmentation to restrict the scope of knowledge binding, so that each piece of knowledge is primarily bound to a limited local context. Specifically, let L denote the segment length in words and W_j the number of words in narrative D_j. We split D_j into $K_j = \lceil W_j/L \rceil$ consecutive, non-overlapping segments

$$D_j \;\rightarrow\; (S_{j,1}, \ldots, S_{j,K_j}),$$

where each segment $S_{j,k}$ contains L consecutive words (the last segment may contain fewer). Segmentation is performed at the word level so that segment boundaries never split a word. Each segment is then tokenized and treated as an independent training unit for subsequent entity-level diffusion. We set $L = 64$ words in our main experiments and study the effect of L in Section 4.4.

3.2.3 Entity-level Diffusion

We inject knowledge from each segment using a masked diffusion language modeling objective (Ye et al., 2025b; Nie et al., 2025). As illustrated in the entity-level diffusion stage of Figure 3, a subset of tokens is randomly masked during training, and a Transformer with bidirectional attention aggregates information from the remaining unmasked context to reconstruct the masked tokens. Unlike autoregressive training, which binds each token to its fixed preceding context, diffusion creates diverse cue-to-knowledge bindings under varying masking patterns, allowing different subsets of the narrative to serve as cues for the same knowledge.

Directly applying diffusion at the token level, however, can introduce undesirable bindings within named entities. Tokens constituting a named entity naturally occur together and represent a single entity. When these tokens are masked independently, part of an entity may remain visible while the rest is masked. The model may consequently learn to reconstruct the remaining entity tokens from the partial entity together with the surrounding knowledge, rather than learning a binding between the complete entity and its associated knowledge. For example, independently masking the entity "Clotilde" may produce "Cl[MASK]ilde," encouraging the model to predict "ot" from the visible fragments "Cl" and "ilde" rather than from the surrounding narrative.

To avoid such partial-entity bindings, we perform masking at the entity level using the named-entity spans identified during Entity Processing. After tokenization, let $\mathbf{d} = (d_1, \ldots, d_n)$ denote the token sequence of a segment $S_{j,k}$, where n is its number of tokens (we omit the subscripts j, k for brevity). For a noise level $\tau \in [\tau_{\min}, \tau_{\max}]$, each named-entity span is masked as a whole with probability τ, while each non-entity token is masked independently with the same probability.

Let $\mathcal{M}_\tau \subseteq \{1, \ldots, n\}$ denote the resulting set of masked positions, and let $\tilde{\mathbf{d}}_\tau$ denote the corrupted segment obtained by replacing the tokens at positions in $\mathcal{M}_\tau$ with `[MASK]`. We optimize the masked diffusion objective:

$$\mathcal{L} = \mathbb{E}_{\tau,\, \mathcal{M}_\tau} \left[-\frac{1}{\tau} \sum_{i \in \mathcal{M}_\tau} \log p_\theta \left(d_i \mid \tilde{\mathbf{d}}_\tau \right) \right], \tag{1}$$

where θ denotes the model parameters and $\tau \sim \mathrm{Uniform}(\tau_{\min}, \tau_{\max})$ is drawn once per segment. Following standard masked diffusion training (Sahoo et al., 2024; Nie et al., 2025), the $1/\tau$ weighting compensates for the varying number of masked positions across noise levels. Because each named-entity span is masked with the same probability τ as non-entity tokens, every token is still masked with marginal probability τ, so the $1/\tau$ weighting remains valid under entity-level masking. This entity-level masking constrains knowledge binding to complete named entities, reducing unnecessary token-level bindings within each entity while retaining the diverse cue-to-knowledge bindings introduced by diffusion.

4 Experiments

4.1 Experimental Setup

Baselines. We compare against five approaches: (1) **Raw-FT**, which fine-tunes on the raw source corpus without any synthetic augmentation; (2) **Rephrase-FT** (Ovadia et al., 2024), which fine-tunes on a rephrased corpus that rewrites the story in different words and style while keeping the same plot, characters, and order; (3) **SynthQA** (Lewis et al., 2021; Mecklenburg et al., 2024), which fine-tunes on question–answer pairs generated from the source, injecting the content as discrete QA points; (4) **EntiGraph** (Yang et al., 2025), which extracts entities from the story and prompts an LLM to describe their pairwise relationships, casting the story as a knowledge-graph-like structure and expanding it into a synthetic corpus; and (5) **ActiveRead** (Lin et al., 2026), which prompts an LLM to generate diverse, document-specific learning strategies and then synthesizes training data by instantiating these strategies.

Datasets. We use two standard narrative QA benchmarks. QuALITY (Pang et al., 2022) contains medium-length stories (4,000–6,000 words), paired with multiple-choice questions. NarrativeQA (Kočiskỳ et al., 2018) provides long books (80,000 words on average) paired with human-authored questions and reference answers. We use all 265 books from QuALITY and a length-stratified random sample of 50 books from NarrativeQA.

Metrics. We evaluate performance on the dataset-paired QA tasks. Specifically, we report multiple-choice QA accuracy on QuALITY and free-form QA accuracy on NarrativeQA. For NarrativeQA, we use Claude Opus 5 (`claude-opus-5`) as an LLM judge to determine whether the candidate answer contains the semantic content of the reference answer, following evidence that LLM-based evaluation is more reliable than lexical-overlap metrics for free-form narrative QA (Bonomo et al., 2025). The exact judge version and decoding settings are reported in Appendix C.

For episodic recall, given a narrative title and an event cue, the model recalls what happens immediately before or after the event. We report Hit Ratio, the fraction of questions for which at least one reference fact is recalled, as judged by Claude Opus 5 (`claude-opus-5`). Full construction and evaluation details are provided in Appendix A, and the judge configuration is reported in Appendix C.

Implementation details. We use Qwen2.5-7B-Instruct as the backbone for all autoregressive baselines and as the generator for methods requiring synthetic augmentation. EntDiff uses Dream-v0-Instruct-7B (Ye et al., 2025b), a diffusion model adapted from Qwen2.5-7B with the same architecture and initial weights. At inference time, we keep the iterative mask-predict decoding of the diffusion model but unmask positions in a fixed left-to-right order, committing the left-most masked token at each step.

For augmentation-based approaches, we generate 455M synthetic tokens on QuALITY and 540M tokens on NarrativeQA and train for one epoch over the generated corpus. EntDiff is trained on the preprocessed source narratives for 50 epochs on QuALITY (75M training tokens) and 40 epochs

on NarrativeQA (216M training tokens), using non-overlapping segments of 64 words. Raw-FT is trained on the source corpus for 10 epochs, as training for more epochs further degraded its performance. For every method, we report results at the final checkpoint. We use three fine-tuning random seeds (42, 1000, and 2000).

Exact model details, optimization hyperparameters, diffusion noise settings, decoding configurations, and software versions are provided in Appendix C.

4.2 Main Results

Table 1: Overall performance on two datasets. Multiple-choice accuracy and QA accuracy measure performance on the dataset-paired QA tasks, while the Hit Ratio measures episodic recall. Values after $\pm$ are standard deviations over three fine-tuning seeds. The best result in each column is shown in bold.

	QuALITY		NarrativeQA	
Method	MC Acc. (%)	Hit Ratio (%)	QA Acc. (%)	Hit Ratio (%)
AR Base	42.01	2.80	14.50	2.71
Diff. Base	41.90	2.60	14.11	2.46
Raw-FT	44.29 $\pm$0.35	2.50 $\pm$0.28	17.95 $\pm$0.55	3.05 $\pm$0.34
Rephrase-FT	49.38 $\pm$0.42	12.04 $\pm$0.46	30.14 $\pm$0.68	12.08 $\pm$0.52
SynthQA	**54.28** $\pm$0.39	8.86 $\pm$0.37	31.92 $\pm$0.63	8.70 $\pm$0.45
EntiGraph	53.18 $\pm$0.46	11.66 $\pm$0.49	31.37 $\pm$0.71	10.43 $\pm$0.50
ActiveRead	54.05 $\pm$0.37	9.44 $\pm$0.41	35.10 $\pm$0.74	7.15 $\pm$0.43
EntDiff (Ours)	54.00 $\pm$0.41	**14.01** $\pm$0.44	**36.71** $\pm$0.69	**14.59** $\pm$0.48

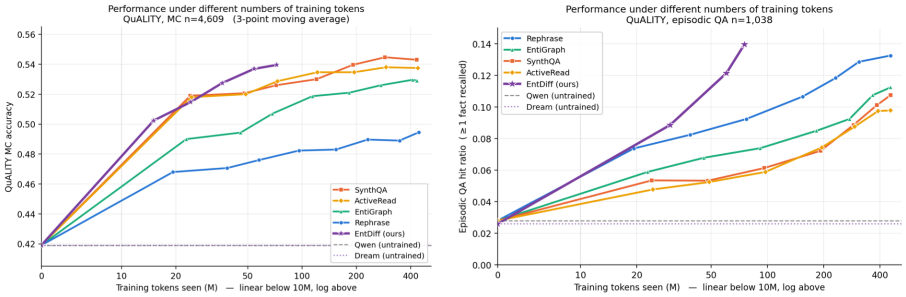
Table 1 compares EntDiff with existing knowledge injection methods on both dataset-paired QA and episodic recall. We first observe that AR Base and Diff. Base achieve similar performance on both datasets, suggesting that adapting Qwen into Dream does not itself inject substantial task-relevant narrative knowledge and supporting the comparability of the two base models. Second, Raw-FT provides only limited improvement over AR Base, particularly on episodic recall. This result further supports our motivation that conventional autoregressive training on long narratives is ineffective at making the injected knowledge reliably accessible.

An interesting observation is that baselines that tend to preserve or reorganize the narrative content rather than explicitly optimize QA ability, such as Rephrase-FT and EntiGraph, achieve relatively stronger episodic recall but weaker performance on the dataset-paired QA tasks. For example, Rephrase-FT reaches Hit Ratios of 12.04% and 12.08% on QuALITY and NarrativeQA, respectively. In contrast, QA-oriented baselines such as SynthQA and ActiveRead achieve stronger dataset-paired QA performance but weaker episodic recall. For instance, ActiveRead achieves 35.10% QA accuracy on NarrativeQA but only a 7.15% Hit Ratio. This suggests that strong dataset-paired QA performance does not necessarily imply strong episodic recall.

EntDiff performs strongly on both dimensions while learning primarily from the source narrative rather than from a task-specific synthetic corpus. It achieves performance comparable to the best baselines on the dataset-paired QA tasks, while obtaining the highest Hit Ratios of 14.01% and 14.59% on QuALITY and NarrativeQA, respectively, consistent with our hypothesis that reducing the binding–access mismatch improves knowledge accessibility. A qualitative comparison of episodic recall across methods is provided in Appendix A.

4.3 Efficiency

Data generation cost. Augmentation-based methods require hundreds of millions of generated training tokens, incurring substantial generation costs. ActiveRead is particularly expensive because it requires two rounds of generation for each document: one to generate learning strategies and another to instantiate them. In contrast, our approach only requires lightweight pronoun resolution as preprocessing, reducing the generation cost by orders of magnitude (Table 2).



(a) QA Accuracy vs. number of training tokens

(b) Episodic Hit Ratio vs. number of training tokens

Figure 4: Performance under different amounts of training data on QuALITY. (a) Multiple-choice QA accuracy and (b) Episodic Hit Ratio as a function of the number of training tokens. Augmentation-based baselines are trained on synthetically generated data, whereas EntDiff is trained on the source corpus after pronoun resolution. Dashed horizontal lines indicate AR Base (Qwen) and Diff. Base (Dream) without fine-tuning.

Table 2: Data generation and training costs. All times are reported in GPU hours on NVIDIA H100-80GB GPUs. The number in parentheses after each dataset name is the token count of its source corpus. For augmentation-based methods, generated tokens form the synthetic training corpus; for EntDiff, generated tokens are used for pronoun resolution and named-entity identification.

		Data Generation		Training	
Dataset (source tokens)	Method	Generated Tokens	GPU-h	Training Tokens	GPU-h
QuALITY (1.5M)	Rephrase-FT	455M	30.8	455M	24.3
	SynthQA	455M	25.7	455M	21.6
	EntiGraph	455M	24.0	455M	21.2
	ActiveRead	455M	67.9	455M	21.1
	EntDiff	22K	0.2	75M	8.1
NarrativeQA (5.4M)	Rephrase-FT	540M	48.4	540M	26.2
	SynthQA	540M	31.2	540M	26.0
	EntiGraph	540M	27.5	540M	26.5
	ActiveRead	540M	75.7	540M	26.3
	EntDiff	155K	0.4	216M	21.1

Learning efficiency. EntDiff is also more data-efficient in terms of training tokens. Due to stochastic masking, each pass presents a different corrupted version of the same narrative, exposing the model to diverse training instances across epochs. Because training is performed primarily on source narrative segments rather than a massively expanded synthetic corpus, the total number of training tokens is substantially smaller. As shown in Figure 4, EntDiff achieves performance comparable to the strongest augmentation-based methods on dataset-paired QA with fewer training tokens, while attaining considerably higher episodic recall. It uses about 6× fewer training tokens on QuALITY and 2.5× fewer on NarrativeQA. The reduction in training time is smaller than the reduction in training tokens because EntDiff trains on shorter segments, resulting in lower token throughput.

4.4 Sensitivity Analysis

Table 3: Performance of EntDiff with different segment lengths L (in words).

	QuALITY		NarrativeQA	
Segment length L	MC Acc. (%)	Hit Ratio (%)	QA Acc. (%)	Hit Ratio (%)
32	53.18	12.76	33.28	13.01
64	**54.00**	**14.01**	**36.71**	**14.59**
128	52.24	13.60	35.40	14.28
512	50.10	10.89	31.27	12.72
2,048	49.11	6.38	24.80	10.54

Table 3 shows that a segment length of $L = 64$ words gives the best dataset-paired QA and episodic recall on both datasets, with performance gradually decreasing as segments become longer. This trend supports our motivation for segmentation: longer segments expose each prediction to a larger amount of surrounding context, making the resulting cues more complex. Such bindings are harder to activate from the short and sparse cues provided at inference time, thereby increasing the binding–access mismatch. However, using overly short segments can also be suboptimal. Reducing the segment length from 64 to 32 words decreases both accuracy and episodic recall, suggesting that sufficient local context is still necessary to capture meaningful relations among entities and events.

4.5 Ablation Study

Table 4: Ablation study of EntDiff on QuALITY and NarrativeQA.

	QuALITY		NarrativeQA	
Variant	MC Acc. (%)	Hit Ratio (%)	QA Acc. (%)	Hit Ratio (%)
EntDiff (full version)	54.00	14.01	36.71	14.59
w/o Pronoun Resolution	53.01	12.64	33.73	11.28
w/o Short Segmentation ($L = 2{,}048$)	49.11	6.38	24.80	10.54
w/o Diffusion (AR)	46.15	6.26	19.21	8.79
w/o Entity-level Masking	53.41	12.83	35.13	12.67

Removing pronoun resolution consistently reduces both dataset-paired QA and episodic recall, indicating the benefit of making ambiguous entity references explicit before knowledge injection. Because full narratives exceed the maximum sequence length of 4,096 tokens, we approximate the setting without segmentation using 2,048-word segments. Removing short segmentation causes a substantially larger degradation on both datasets, since overly broad contexts make learned knowledge harder to access from short inference-time cues. Replacing entity-level masking with independent token-level masking also reduces performance, showing that treating named entities as complete units improves knowledge binding. Replacing diffusion training with autoregressive fine-tuning causes the largest overall degradation, highlighting the importance of relaxing the fixed left-to-right binding constraint.

5 Limitations

EntDiff focuses on knowledge explicitly present in the source text and does not directly introduce knowledge that must be derived through explicit reasoning, such as implicit relations, inferred facts, or higher-level insights. Lightweight synthetic or reasoning-based augmentation could complement EntDiff while preserving much of its data efficiency.

6 Conclusion

We study data-efficient narrative knowledge injection and identify the *binding–access mismatch* as a key challenge in learning directly from narrative corpora: knowledge acquired under specific training contexts may remain difficult to access from different inference-time cues. To address this problem, we propose EntDiff, which reshapes how narrative knowledge is bound during training and enables more flexible access to the learned knowledge. Experiments show that EntDiff matches or outperforms strong augmentation-based methods on dataset-paired QA while substantially improving episodic recall, generating more than three orders of magnitude fewer tokens during data preparation and training on 2.5–6$\times$ fewer tokens. These results suggest that effective knowledge injection need not rely on exhaustive synthetic augmentation; improving how knowledge is bound and accessed provides a promising alternative for learning directly from source corpora.

7 Reproducibility Statement

We describe the model architecture and training objective in Section 3, the datasets, baselines, evaluation metrics, and core training setup in Section 4.1, and the complete implementation and eval-

uation configuration in Appendix C. The exact prompts used for episodic recall and NarrativeQA evaluation are provided in Appendices A and B, respectively. We provide anonymized code and configuration files at `https://anonymous.4open.science/r/EntDiff-E4BC`.

8 AI USE STATEMENT

Qwen2.5-7B-Instruct was used for synthetic-data generation and lightweight preprocessing, including pronoun resolution and named-entity categorization. Claude Opus 5 was used for LLM-based evaluation of episodic recall and NarrativeQA. Generative AI tools were additionally used for language polishing and figure refinement. All AI-assisted outputs were reviewed by the authors, who take full responsibility for the final manuscript.

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 1067–1077. PMLR, 21–27 Jul 2024. URL `https://proceedings.mlr.press/v235/allen-zhu24a.html`.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021.

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Stickland, Tomek Korbak, and Owain Evans. The reversal curse: Llms trained on "a is b" fail to learn "b is a". In *International Conference on Learning Representations*, 2024.

Tommaso Bonomo, Luca Gioffré, and Roberto Navigli. Literaryqa: Towards effective evaluation of long-document narrative qa. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 34074–34095, 2025.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models. In *International Conference on Learning Representations*, 2025.

Minki Kang, Sung Ju Hwang, Gibbeum Lee, and Jaewoong Cho. Latent paraphrasing: Perturbation on layers improves knowledge injection in language models. In *Advances in Neural Information Processing Systems*, 2024.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115, 2021.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*, 2022.

Jessy Lin, Vincent-Pierre Berges, Xilun Chen, Wen-Tau Yih, Gargi Ghosh, and Barlas Oğuz. Learning facts at scale with active reading. In *The Fourteenth International Conference on Learning Representations*, 2026.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2024.

Nick Mecklenburg, Yiyou Lin, Xiaoxiao Li, Daniel Holstein, Leonardo Nunes, Sara Malvar, Bruno Silva, Ranveer Chandra, Vijay Aski, Pavan Kumar Reddy Yannam, Tolga Aktas, and Todd Hendry. Injecting new knowledge into large language models via supervised fine-tuning. *arXiv preprint arXiv:2404.00213*, 2024.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. Fine-tuning or retrieval? comparing knowledge injection in llms. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 237–250, 2024.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, et al. Quality: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5336–5358, 2022.

Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. Diffusion beats autoregressive in data-constrained settings. *arXiv preprint arXiv:2507.15857*, 2025.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, 2024.

Kexian Tang, Jiani Wang, Shaowen Wang, and Kaifeng Lyu. Spa: A simple but tough-to-beat baseline for knowledge injection. *arXiv preprint arXiv:2603.22213*, 2026.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Endel Tulving. Elements of episodic memory. 1983.

Endel Tulving et al. Episodic and semantic memory. *Organization of memory*, 1(381-403):1, 1972.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candes, and Tatsunori Hashimoto. Synthetic continued pretraining. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=07yvxWDSla.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. In *International Conference on Learning Representations*, 2025a.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025b.

Adam Zweiger, Jyothish Pari, Han Guo, Ekin Akyürek, Yoon Kim, and Pulkit Agrawal. Self-adapting language models. In *Advances in Neural Information Processing Systems*, 2025.

A Episodic Recall Evaluation

Our episodic recall evaluation measures whether a model can reconstruct the local narrative context before or after a specific event, rather than only answering semantic questions about the document.

Cue Generation. We select sentences that can serve as reliable episodic retrieval cues. Each cue is a sentence that describes a unique and specific event involving identifiable entities at a particular moment in the narrative. We use Claude Opus 5 (`claude-opus-5`) to determine whether a sentence can be expressed in the form "at this moment, X did, said, noticed, or thought Y," and discard sentences describing habits, background information, general descriptions, or author commentary.

We sample cue sentences throughout each document rather than concentrating them near the beginning. Each sentence is used for only one question, and we generate the same number of questions for every document so that longer documents do not receive disproportionate weight in the final score.

Reference and Fact Construction. For each selected cue, we extract the passage immediately before or after it, extending outward one complete sentence at a time until reaching approximately 80 words. We stop only at sentence boundaries to avoid creating reference passages that contain truncated events. The resulting passage serves as the reference episode.

We then use Claude Opus 5 (`claude-opus-5`) to decompose each reference passage into a set of atomic episodic facts. Each fact describes one event, such as an entity performing an action, saying something, noticing something, or experiencing an event. Entities are referred to using the names appearing in the passage, and purely descriptive content is excluded. Most reference passages contain between three and seven facts. The final evaluation set contains 1,038 questions and 5,955 reference facts for QuALITY, and 1,035 questions and 5,517 reference facts for NarrativeQA, averaging 5.7 and 5.3 facts per question respectively.

Evaluation. For each question, the model is given the narrative title and the cue sentence and is asked to recall what happens before or after the cue. The LLM judge, Claude Opus 5 (`claude-opus-5`), then evaluates each reference fact independently by determining whether that fact is present in the generated answer. We report the Hit Ratio, defined as the fraction of questions for which the generated answer contains at least one reference fact:

$$\text{Hit Ratio} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbb{I}\left[\exists f \in \mathcal{F}_q : \ f \text{ is recalled}\right], \tag{2}$$

where $\mathcal{Q}$ denotes the set of evaluation questions and $\mathcal{F}_q$ denotes the set of reference facts for question q. Compared with directly assessing the overall agreement between a generated passage and the reference passage, fact-level verification provides a more consistent evaluation criterion by reducing each judgment to whether a specific reference fact is present in the generated answer.

Below we provide two examples of the episodic recall evaluation.

Examples of Episodic Recall

Example 1: *from "And Then the Town Took Off" by Richard Wilson*

Cue sentence. A ham radio operator reported an unidentified voice as saying plaintively: "Cold up here!"

Reference passage (preceding the sentence). The object loomed out of a cloudbank at twelve thousand feet and Studley changed course to avoid it. He noted with only minimum satisfaction that his co-pilot also saw the thing and wondered why it wasn't moving at the terrific speed flying saucers were allegedly capable of. Then he saw the church steeple on it. A few minutes later he had relayed a message from Superior, formerly of Ohio, addressed to whom it might concern: It said that Superior had seceded from Earth. One other radio message came from Superior, now airborne, on that first day.

Facts.

1. The object looms out of a cloudbank at twelve thousand feet and Studley changes course to avoid it.
2. Studley notes his co-pilot also sees it and wonders why it isn't moving at the speed flying saucers supposedly reach.
3. Studley sees a church steeple on the object.
4. A few minutes later he relays a message from Superior saying it had seceded from Earth.
5. Only one other radio message came from the airborne Superior that first day.

Example 2: *from "Breakaway" by Stanley Gimble*

Cue sentence. Phil turned the car off the highway onto the rutted dirt road that led across the sand to the field where the ship waited.

Reference passage (following the sentence). In the distance they could see the beams of the searchlights as they played across the take-off zone and swept along the top of the high wire fence stretching out of sight to right and left. At the gate they were stopped by the guard. He read Phil's pass, shined his flashlight in their faces, and then saluted. "Good luck, colonel," he

said, and shook Phil's hand. "Thanks, sergeant. I'll be seeing you next week," Phil said, and smiled. They drove between the rows of wooden buildings that lined the field, and he parked near the low barbed fence ringing the take-off zone.

Facts.

1. They see searchlight beams playing across the take-off zone and the high wire fence.
2. The guard stops them at the gate.
3. The guard reads Phil's pass, shines his flashlight in their faces, and salutes.
4. The guard wishes Phil luck and shakes his hand.
5. Phil thanks the sergeant, says he'll see him next week, and smiles.
6. They drive between rows of wooden buildings and Phil parks near the low barbed fence around the take-off zone.

Qualitative Model Comparison. The following example compares the episodic recall of different knowledge-injection methods for the same retrieval cue. Bold text marks generated content that matches a reference fact.

Example of Episodic Recall

Question: In the story *In Case of Fire* by Randall Garrett, what happens after "Bobby did not need to hear Pop's reply to know that it was. His swift intake of breath was enough, the shine in his eyes as he peered out the observation port"?

Reference Facts: 1. Pop says "Eros!"; 2. Bobby sees a small clean sphere spinning lazily like a forgotten top; 3. The planetoid appears silver, green and rusty brown, faintly blued by distance; 4. Sunlight reflects from seas covering half its surface, and mountains cast long shadows on the plains; 5. Bobby thinks it is not a bad looking little place.

Rephrase-FT: The captain, Gwayne, turned to face the newcomer with a frown, noting the man's weary appearance and the dark circles under his eyes. He had been awake for over twenty hours, having spent much of the time overseeing the landing operations and dealing with the unpredictable behavior of the crew. The ship's interior was dimly lit, with the only light coming from the glowing control panels, casting long shadows across the room.

EntiGraph: He saw the planetoid ahead. It wasn't big; it wasn't even very small. But it was lovely, all right, in an odd sort of way. He could see **the water now, a vast sheet of water**, almost as smooth as a rug, and running right up to the base of the ship. The water was still, except for the tiny ripples that traveled out under the impetus of the ship's descent. There was no sound, except the thudding of his own heart, and the faint hiss of escaping air from somewhere nearby in the ship.

ActiveRead: The beach was a narrow strip of sand dunes backed by an almost impenetrable wall of rock. The ship struck at 1300 hours and three minutes, and Pop had been right; it did settle a bit more. But not much. In another minute they were all dressed and heading up the slope toward the broad flat-topped ridge of rock that separated the two halves of the landing field.

EntDiff: Eros was waiting for them. A tiny, round world, gleaming in the morning sunlight, a topaz set in black velvet. A world that beckoned, a world that beckoned. For a long time they watched it, and it watched them. Then, as the ship nosed ever downward, the surface detail became clear. **Silver-green plains, striped with dark brown and black, the faint blue of water in a shallow, shallow sea; and, in the distance, the crags and peaks of a mountain range**. Below them, the great, fluffy fluffy white clouds that hid the surface from view.

Below we provide the prompt used for episodic recall evaluation.

Prompt for Episodic Recall Evaluation

A model was given an event from a story and asked to recall what happened before or after that event. You are evaluating whether each reference fact is recalled in the model's prediction.
Each item contains:

`SENT`: the event used as the retrieval cue
`FACTS`: a numbered list of reference facts
`PRED`: the model-generated answer

For each reference fact, assign one label:

- **H**: The fact is present in `PRED`. Judge by meaning; paraphrases and semantically equivalent descriptions count.
- **M**: The fact is not present in `PRED`, including when the prediction gives incorrect or unrelated information.

Evaluate each fact independently. Ignore additional information in `PRED` that is unrelated to the reference facts. Reply with one line of JSON per item, in input order, and nothing else. The `labels` field contains one label for each reference fact:

```
{"uid":"...","labels":"MHHMM"}
```

A question is counted as a hit if at least one of its reference facts is labeled `H`. The Hit Ratio is the proportion of questions that are hits.

B NarrativeQA Evaluation

NarrativeQA requires free-form answer generation, for which lexical-overlap metrics such as Exact Match, token-level F1, and ROUGE-L may poorly reflect semantic correctness. Following the findings of LiteraryQA (Bonomo et al., 2025), we therefore use an LLM judge to determine whether each candidate answer contains the semantic content of the reference answer. Specifically, we use Claude Opus 5 (`claude-opus-5`) as the judge and compute accuracy using the following prompt. The exact model version and decoding configuration are reported in Appendix C:

Prompt for NarrativeQA Answer Evaluation

 # Grade NarrativeQA free-form answers — OUTPUT 1 OR 0

You are checking whether the information in the reference answer is contained in the candidate answer. This is not a quality check. A candidate may be longer than the reference, contain surrounding narration, or use different wording. None of these should be penalized as long as the reference answer's meaning is present.

The rule

Output `1` if the reference answer's content is present in the candidate answer. The following still count as correct:

- The candidate is much longer than the reference or contains surrounding narration.
- The candidate includes additional facts or context.
- The candidate paraphrases the reference rather than using the same wording.
- The candidate uses an alias, title, or another name for the same entity.
- The candidate expresses the same fact using semantically equivalent wording.
- The candidate is truncated after the relevant answer has already appeared.

Output `0` if the reference answer's content is not present, is contradicted by the candidate, or the candidate is too vague to establish the answer.

Output

For each item, output only: 1 or 0.

C Reproducibility Details

This appendix records the implementation choices required to reproduce the experiments.

Model and preprocessing configuration. The backbones, preprocessing and evaluation models, and data-sampling settings are listed below.

Table 5: Model, preprocessing, and data-sampling settings.

Item	Setting
Autoregressive backbone	`Qwen/Qwen2.5-7B-Instruct`
Diffusion backbone	`Dream-org/Dream-v0-Instruct-7B`
Pronoun resolution	`Qwen/Qwen2.5-7B-Instruct`. Candidate pronouns are located by regex and indexed; the model returns only `index=name` or `SKIP`, rather than rewriting the narrative.
Named entity identification	Candidate entity strings are extracted with deterministic capitalization-based rules and categorized by Qwen2.5-7B-Instruct. We retain {`PERSON`, `PLACE`, `ORG`, `BOOK_ENTITY`} and map retained mentions to token spans for entity-level masking.
Atomic-fact decomposition model	Claude Opus 5 (`claude-opus-5`)
Episodic recall / NarrativeQA judge	Claude Opus 5 (`claude-opus-5`)
NarrativeQA sampled-book seed	`42`
NarrativeQA book sampling	Length-stratified random sample of 50 NarrativeQA books, 10 per stratum (`S1_5k`, `S2_17k`, `S3_55k`, `S4_104k`, `S5_212k`).

Optimization settings. Unless otherwise stated, the following settings are used for fine-tuning.

Table 6: Optimization settings used for knowledge injection.

Hyperparameter	Setting
Fine-tuning scope	Full-parameter (no PEFT)
Optimizer	AdamW (`adamw_torch`, HuggingFace default: $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$)
Learning rate	2×10^{-5}
Learning-rate schedule	`constant_with_warmup`
Warmup	500 steps
Weight decay	0.01
Gradient clipping	1.0 (HuggingFace default; not overridden)
Per-GPU batch size	4
Gradient accumulation steps	4
Effective batch size	16 sequences
Maximum sequence length	4096 tokens
Training precision	Pure bfloat16: model weights, gradients, and AdamW states are all stored in bf16 (no fp32 master weights)
Training random seeds	Three seeds: `42`, `1000`, and `2000`
GPUs	1 NVIDIA H100-80GB (single-GPU, no tensor/data parallelism)
Checkpoint selection	Each method is trained for a fixed, pre-specified number of epochs (Section 4.1) and evaluated at the final checkpoint.

EntDiff diffusion settings. EntDiff uses non-overlapping 64-word segments in the main experiments. After tokenization, named-entity spans are masked jointly, while non-entity tokens are masked independently. The remaining diffusion-specific settings are:

Table 7: Diffusion-specific settings for EntDiff.

Hyperparameter	Setting
Noise-level distribution $p(\tau)$	$\tau \sim \mathrm{Uniform}(\tau_{\min}, \tau_{\max})$, drawn once per segment (not per token)
$\tau_{\min}$	0.05
$\tau_{\max}$	0.95
Loss normalization	Cross-entropy over masked positions only (-100 elsewhere), weighted by $1/\tau$ as in Eq. (1), summed within a sequence and averaged over the batch.

Synthetic-data generation. All augmentation-based corpora are generated with Qwen2.5-7B-Instruct using vLLM with prefix caching. The exact generation settings are:

Table 8: Generation settings for augmentation-based baselines and preprocessing.

Generation parameter	Setting
Temperature	1.0
Top-p	1.0
Top-k	disabled
Maximum new tokens	4096
Generation seed	42
Prefix caching	Enabled.
vLLM engine	`tensor_parallel_size`=1, `gpu_memory_utilization`=0.9, `max_model_len` unset

Inference and evaluation settings. The exact evaluation prompts for episodic recall and NarrativeQA are given in Appendices A and B. Generation and judge settings are listed below.

Table 9: Inference and evaluation settings.

Evaluation parameter	Setting
Generation decoding	Greedy. The AR models use `do_sample=False`; the diffusion model uses iterative mask-predict decoding with a left-to-right unmasking order (one token per step) at temperature 0.
Generation temperature / top-p	0 / not used (greedy)
Maximum new tokens	160 for all models
Generation seed	Not applicable (greedy decoding is deterministic)
Judge interface	Claude CLI
Judge model	Claude Opus 5 (`claude-opus-5`)
Judge sampling configuration	No temperature, top-p, or top-k parameters are explicitly specified; the default Claude Opus 5 decoding configuration is used
Thinking configuration	No explicit thinking configuration is specified; the default Claude Opus 5 setting is used
Judge maximum tokens	Not set; CLI default. A 1800 s per-call timeout is applied
Judge runs per example	1.
QuALITY answer-selection	Zero-shot single-token selection. The question and options are wrapped in the chat template; the diffusion model appends one `<mask>` at the assistant's first output position and takes the argmax there, while the AR model takes the argmax of the next-token distribution at the final prompt position. The decoded token is matched against `A`–`D`.

Software environment. PyTorch 2.11.0 (`+cu130`); Transformers 4.57.6; TRL 0.29.0; Datasets 4.5.0; vLLM 0.22.1; CUDA 13.0. Jobs run on a Slurm cluster with one NVIDIA H100-80GB per job.