

Think Before you Write: QA-Guided Reasoning for Character Descriptions in Books

Argyrios Papoudakis Mirella Lapata Frank Keller

Institute of Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
a.papoudakis@sms.ed.ac.uk, {mlap, keller}@inf.ed.ac.uk

Abstract

Character description generation is an important capability for narrative-focused applications such as summarization, story analysis, and character-driven simulations. However, generating accurate character descriptions from long-form narratives (e.g., novels) is challenging: models must track evolving attributes (e.g., relationships and events), integrate evidence scattered across the text, and infer implicit details. Despite the success of reasoning-enabled LLMs on many benchmarks, we find that for character description generation their performance improves when built-in reasoning is disabled (i.e., an empty reasoning trace). Motivated by this, we propose a training framework that decouples reasoning from generation. Our approach, which can be applied on top of long-context LLMs or chunk-based methods, consists of a reasoning model that produces a structured QA reasoning trace and a generation model that conditions on this trace to produce the final character description. Experiments on two datasets (BookWorm and CroSS) show that QA-guided reasoning improves faithfulness, informativeness, and grounding over strong long-context baselines.¹

1 Introduction

Writers craft characters to create engaging stories that invite readers to experience actions, emotions, and goals from the characters’ perspectives. Characters form the core of the narrative, with other elements such as plot, conflict, setting, and theme, built around them. A growing body of work in natural language processing aims to model characters for story analysis (Zhu et al., 2023), narrative generation and summarization (Fan et al., 2018), and persona simulation within character-driven interactions (Shao et al., 2023). Yet despite their importance, characters remain difficult to model computationally, particularly in long-form narratives where their traits, motivations, and relationships evolve over extended spans of text and through complex and often non-linear plots (Chaturvedi et al., 2015; Vishnubhotla et al., 2024).

¹Code and data are available at <https://github.com/apapoudakis/qa-guided-reasoning>

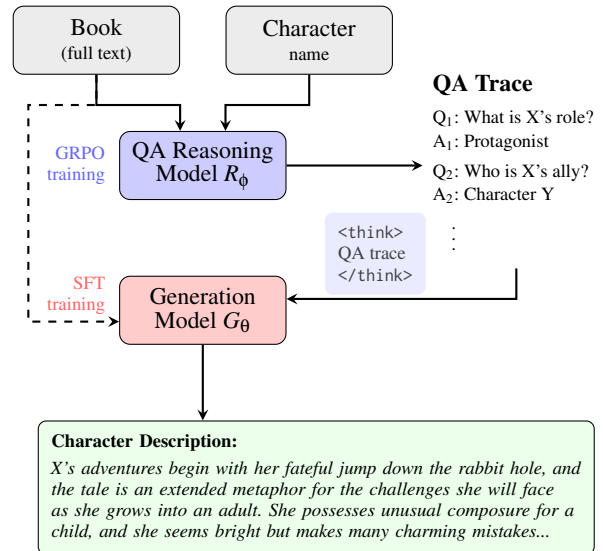


Figure 1: Overview of QA-based guided reasoning for character description generation. Given a book and character name, the *reasoning model* generates a structured QA trace capturing salient character information. This trace is injected into the thinking tokens of the *generation model*, which produces the final character description. The reasoning model can be trained with GRPO and the generation model with SFT.

tionally, particularly in long-form narratives where their traits, motivations, and relationships evolve over extended spans of text and through complex and often non-linear plots (Chaturvedi et al., 2015; Vishnubhotla et al., 2024).

Consequently, a central difficulty in character-centric understanding stems from the sheer length of the input. To address this, earlier work adopted chunk-based approaches using LLMs with short-context abilities to process stories either hierarchically (Wu et al., 2021) or incrementally (Chang et al., 2023). While these methods are computationally efficient, they often struggle to capture all relevant information within individual chunks and to integrate evidence reliably across chunks. Retrieval-augmented methods have also been proposed to select salient context (Xu et al., 2023);

however, the retrieved passages are frequently disjoint and may omit critical details, undermining global narrative coherence. More recently, long-context LLMs, capable of processing up to millions of tokens (Team et al., 2024), have been used to analyze stories in a single pass. Nevertheless, even these models face known limitations, including position bias (Liu et al., 2024), difficulty in exploiting in-context examples (Li et al., 2024) and challenges in integrating information across distant spans (Tian et al., 2025).

Beyond context scaling, long narratives present an additional challenge: much information is implied rather than explicitly stated. Models must therefore reason over long inputs (>100k tokens), integrating evidence from different text passages. This requires inferring implicit details, combining information which is being revealed piecemeal, and filtering out irrelevant content. Recent work has enabled explicit reasoning via post-training reinforcement learning (Shao et al., 2024), achieving strong results on tasks such as mathematics (Yang et al., 2024) and short-form question answering (Rein et al., 2024). However, extending these approaches to non-verifiable settings like text generation remains challenging, as reward computation is difficult when many valid outputs exist across multiple evaluation dimensions.

We address these limitations by focusing on character description generation in books (Brahman et al., 2021), where models must produce factual descriptions given a character and the full story. We propose a two-stage training framework that decouples reasoning from generation: (1) a *reasoning model* produces an explicit reasoning trace, and (2) a *generation model* conditions on this trace to produce the final description (see Figure 1). Importantly, our reasoning model uses question-answer pairs — which have been previously shown to provide a useful abstraction for isolating salient content in summarization (Huot et al., 2023; Narayan et al., 2023; Liu et al., 2025) — as a scaffold to *guide* downstream description generation, forcing the model to “reason through” salient evidence *before* writing. As our reasoning model is input agnostic, it can be integrated on top of different long-context methods, including hierarchical merging, incremental updating, and direct processing with long-context LLMs.

To train the reasoning model, we use Group Relative Policy Optimization (GRPO) (Shao et al., 2024) directly on *gold-standard* character descrip-

tions and *silver-standard* reasoning traces derived from them, thereby avoiding the need to simulate rewards based on the final generated outputs. The generation model can either be trained with supervised fine-tuning (SFT) on these reasoning traces or applied in zero-shot mode.

We evaluate our approach on two character understanding datasets: BookWorm (Papoudakis et al., 2024), which contains books of various genres (primarily novels and plays) from Project Gutenberg, and CroSS (Yuan et al., 2024), which comprises recently published novels. Both datasets pair full-text books with character descriptions. Our experiments reveal that enabling the built-in reasoning capabilities of current LLMs actually degrades performance compared to using an empty reasoning trace. In contrast, our approach effectively improves the faithfulness and informativeness of character descriptions across both datasets, enhancing the performance of existing long-context methods. In summary, we make the following contributions:

- We show that the built-in reasoning mode of current LLMs *reduces* faithfulness for character description generation, with empty traces yielding more accurate outputs.
- We introduce QA-based guided reasoning, a modular approach that decouples reasoning from generation, producing a structured QA trace and then generating the final character description conditioned on that trace.
- We propose a training strategy that optimizes the reasoning model with GRPO using gold-standard descriptions and automatically generated QA traces, avoiding reward design over open-ended generated descriptions.
- We show that guided reasoning yields consistent improvements in faithfulness and informativeness across two datasets and can be integrated with various long-context strategies.

2 Related Work

Character Understanding Work in computational narrative understanding has focused on characters to support story summarization (Zhang et al., 2019), generation (Liu et al., 2020), analysis (Labatut and Bost, 2019), and the simulation of interactions among different personas (Shao et al., 2023). Existing studies have examined multiple character dimensions, including roles (Skowron et al., 2016), relationships (Iyyer et al., 2016), personality (Bamman et al., 2013), and emotions (Kim and Klinger,

2019). More recently, attention has shifted to characters in longer-form narratives (e.g., screenplays, books), where increased complexity (e.g., longer plots, evolving relationships) raises modeling challenges and motivates evaluation of long-context systems. In this setting, several prediction-based tasks have been introduced, such as personality prediction (Sang et al., 2022; Yu et al., 2023) and assigning characters to tropes in scripts (Baruah and Narayanan, 2025). Previous work has also explored richer character representations, e.g., by constructing character sheets (Gurung and Lapata, 2024) or learning character embeddings (Inoue et al., 2022) from books. Another line of research addresses character-based coreference resolution in novels (Martinelli et al., 2025) and screenplays (Baruah and Narayanan, 2023).

In parallel, recent work has framed character understanding as a text-generation problem, producing character descriptions and analyses (Brahman et al., 2021; Papoudakis et al., 2024), structured profiles (Yuan et al., 2024), and persona-conditioned dialogues (Chen et al., 2023). We adopt a similar setting, focusing on free-form character description generation in book-length narratives. Unlike Yuan et al. (2024), we target unstructured descriptions, as existing datasets provide human-written text requiring no additional processing.

Long-context Models and Reasoning Language models can now process extremely long sequences, up to millions of tokens (Team et al., 2024), thanks to advances in sparse attention (Beltagy et al., 2020), long-context post-training (Gao et al., 2025), scalable positional embeddings (Ding et al., 2024), and system (Kwon et al., 2023) and hardware-level (Dao et al., 2022) optimizations. However, greater context length does not reliably translate into better long-document reasoning; recent studies report persistent failure modes, such as position bias (Liu et al., 2024) and difficulty exploiting in-context examples (Li et al., 2024). As the training and inference costs of long-context modeling remain a major barrier (Fu, 2024), previous work has also explored efficient alternatives that handle long documents with short-context models by incorporating retrieval, hierarchical processing, or incremental updating (Chang et al., 2023). By operating on chunks or selectively attending to limited context, these methods can be competitive across a range of long-context settings (Xu et al., 2023). In this paper, we show how guided reasoning can

be layered on top of both chunk-based approaches (retrieval, hierarchical processing, and incremental updating) and long-context LLMs.

Effective reasoning is also crucial for inferring implicit information, filtering irrelevant content, and integrating evidence across long texts. Early work elicited reasoning via prompting (Wei et al., 2022) or fine-tuning (Wei et al., 2021). More recently, post-training with reinforcement learning has become a dominant paradigm, improving performance across a range of tasks (Ouyang et al., 2022), such as mathematics (Uesato et al., 2022) and coding (Shojaee et al., 2023). However, training reasoning models for non-verifiable tasks (e.g., long-form QA and summarization) remains underexplored: reward design is challenging when many outputs are acceptable and quality depends on multiple criteria. Recent work addresses this using BLEU (Chang et al., 2025), perplexity completion (Gurung and Lapata, 2025), or self-certainty (Zhao et al., 2025) to train without external verifiers.

In contrast, we decouple reasoning from generation using a separate reasoning model whose output trace is injected into the generator. This allows us to optimize the reasoner directly with reinforcement learning on its own outputs, without defining rewards over the downstream generated text. Inspired by work showing QA pairs serve as effective *planning proxies* for summarization (Narayan et al., 2023), we use structured QA traces to guide generation. However, our QA pairs function as *reasoning scaffolds* that consolidate distributed evidence about characters, rather than as ordering constraints for generation.

3 QA-Guided Reasoning

We propose a modular approach that decouples reasoning from surface realization for character description generation. Given a book (or long narrative) x and target character c , a *reasoning model* R_ϕ produces an intermediate QA-based reasoning trace T , and a *generation model* G_θ conditions on T to produce a free-form character description \hat{y} . The key idea is to represent intermediate reasoning as *typed* question-answer items that explicitly surface salient character facts before the final description is written. Figure 1 illustrates this architecture; we provide details below.

3.1 QA Reasoning

Let x denote a story and c a target character. When x exceeds the context length of the reasoning model, we split it into m chunks $x = [\chi_1, \dots, \chi_m]$. Given chunk χ_i , the reasoning model R_ϕ outputs either None (if c is not mentioned) or a set of QA tuples:

$$T_i = R_\phi(c, \chi_i) = \{(q_j, e_j, a_j, t_j)\}_{j=1}^{n_i}, \quad (1)$$

where q_j is a question about c , e_j is a short supporting explanation (1–2 sentences), a_j is a short answer (typically 1–4 words), and $t_j \in \mathcal{T}$ is a question type. Following previous work (Papoudakis et al., 2024), we use the question types *role*, *relationship*, *personality*, *event* and *other*, which encourage the model to generate questions with diverse topics without focusing only on specific aspects (see Appendix B). We do not set a maximum number of questions that the model has to generate, but instead ask it to include all the relevant information for the character of interest (or None otherwise). The final reasoning trace is the concatenation of chunk-level traces, $T = \text{CONCAT}(T_1, \dots, T_m)$.

We experimented with several question-generation strategies (see Appendix B), including generating questions directly from the full context, generating questions per chunk and then concatenating them, or multi-step pipelines that first extract topics (Noorbakhsh et al., 2025) or create plans (Li and Zhang, 2024). Since these variants performed similarly or worse, we adopt chunk-based QA generation for simplicity and to facilitate the training of R_ϕ (Section 3.3).

3.2 Generation Model

The generation model G_θ produces a free-form character description conditioned on both the input context and the reasoning trace:

$$\hat{y} = G_\theta(c, x, T). \quad (2)$$

Operationally, we inject T into the model input between special thinking tokens (e.g., `<think> ... </think>`), and prompt the model to return a single-paragraph description. This design is compatible with both reasoning-enabled LLMs and standard instruction-tuned models that do not explicitly emit reasoning traces.

3.3 Training

We finetune the R_ϕ and G_θ separately. Since we do not have gold-standard reasoning traces, we derive

silver-standard supervision from the gold description y by extracting a reference set of QA pairs P . The structured format of these reasoning traces allows us to parse the generated QA-pairs, verify each of them, and provide a combined reward for the reasoning trace of each chunk. We employ an LLM-as-a-judge model to compute precision, the percentage of generated QA-pairs that can be verified from the gold-standard description, and recall, the percentage of gold questions (extracted from descriptions) that the generated QA-pairs can verify. We use the F1 score as a reward signal. For each chunk χ_i , we compute precision and recall as follows:

$$R_{\text{precision}} = \frac{1}{|T_i|} \sum_{(q_j, a_j) \in T_i} (\text{VERIFY}(q_j, a_j, y)) \quad (3)$$

$$R_{\text{recall}} = \frac{1}{|P|} \sum_{(q_j, a_j) \in P} (\text{VERIFY}(q_j, a_j, T_i)) \quad (4)$$

We optimize R_ϕ with GRPO (Shao et al., 2024), which samples multiple traces per prompt and performs a relative policy update without a critic, reducing computational cost.

While we use G_θ in a zero-shot setting in several of our experiments, the generator can also be *fine-tuned* to better exploit guided-QA traces. Concretely, we perform SFT on tuples (c, x, T, y) where T is produced by the guided-QA reasoner (either zero-shot or GRPO-trained), and optimize $\log p_\theta(y | c, x, T)$, optionally applying loss on the injected trace tokens as well. At inference time, the reasoning trace is always supplied by R_ϕ , and G_θ generates only the final description.

3.4 Integrating Guided Reasoning with Long-Context Methods

Because R_ϕ operates on chunks, our QA-guided pipeline is model- and strategy-agnostic: it can be combined with several long-context processing methods by producing traces T_i for the same chunks χ_i that these methods already use. Concretely, we investigate the following settings:

Retrieval-based Methods (Papoudakis et al., 2024) first select a character-relevant subset of the input (e.g., paragraphs mentioning c), yielding $x' \subseteq x$. We simply run R_ϕ over chunks of x' and condition the generator on the resulting trace T and retrieved context.

Hierarchical Methods (Chang et al., 2023) split the input into chunks and generate intermediate descriptions for each chunk, which are subsequently

	BookWorm	CroSS
Books	324	126
Characters	9.74	6.56
Samples	5,869	824
Input Length	97,685	129,113
Output Length	88.79	295.27

Table 1: Dataset statistics: unique books, average characters per book, total examples (book–description pairs), and average input/output length in words.

merged by a second-stage model. We apply guided reasoning only at the first level: for each chunk χ_i , we generate a trace T_i and produce an intermediate description conditioned on (χ_i, T_i) . Merging then operates over the intermediate descriptions without additional reasoning. We employ the reasoning model in a zero-shot setting and trained with GRPO, while the generation model operates in zero-shot mode. We do not finetune the generation model via SFT, as intermediate descriptions for individual chunks are not available, and the model must both generate and merge descriptions, which makes fine-tuning impractical.

Incremental Methods (Chang et al., 2023) process the input chunk-by-chunk, while maintaining a running global description. At step i , the generation model updates the current description using the new chunk and its trace, i.e., $\hat{y}_i = G_\theta(c, \chi_i, T_i, \hat{y}_{i-1})$. This allows newly observed evidence about the character to be integrated as it appears in the narrative. Similar to the hierarchical approach, we only train the reasoning model with GRPO, while the generation model operates in zero-shot mode.

Long-context LLMs can consume the full input x (within their context window) for generation. In this case, we still produce traces chunk-wise with R_ϕ and inject their concatenation T into the long-context generator, i.e., $\hat{y} = G_\theta(c, x, T)$.

4 Experimental Setting

Datasets We use the BookWorm dataset (Papoudakis et al., 2024), which contains books of various genres (mostly novels and plays) from Project Gutenberg paired with character descriptions from literature websites.² For out-of-distribution evaluation, we use CroSS (Yuan et al., 2024), a dataset of novels published in 2022–2023 paired with char-

acter descriptions. Dataset statistics are in Table 1, examples in Appendix C.

Model Comparisons We use Qwen-3-8B (Yang et al., 2025) as the backbone model for all our experiments. We report a *No Context* baseline, where the model is prompted to generate a character description given the character name and book title (i.e., without any story text). This baseline estimates how much parametric knowledge the model already has about the character, serving as a proxy for potential contamination. We also include a *Lead* baseline, truncating the input story to the maximum length supported by the backbone model.

We evaluate our QA-guided approach on top of retrieval-augmented methods that select character-relevant evidence and condition generation on the retrieved context. We consider two strategies: (a) *Coref* runs a coreference resolver to identify chunks in which the target character is mentioned, concatenates the selected chunks, and truncates if the resulting context exceeds the model’s input budget; we use BookNLP³ for coreference resolution, which has been shown to outperform news-trained baselines on literary text (Bamman et al., 2020); (b) *BM25* uses the character name as a query and retrieves chunks by BM25 score which we concatenate (in their order of appearance in the book) the model’s context limit is reached (32k for our experiments with Qwen-3-8b). For both retrieval approaches, we use 512-token length chunks.

We further integrate QA-guided reasoning with two chunk-based long-context strategies, viz., *Hierarchical* processing and *Incremental* updating (see Section 3.4). For both methods, we use 16k-token chunks. Additional implementation details and results are in Appendix A and B.

All methods are evaluated in three settings: without reasoning (empty trace), with the model’s built-in reasoning, and with our proposed guided-QA trace. For guided-QA, we compare zero-shot and GRPO-optimized versions. For retrieval-based methods, we also train an SFT variant, computing the loss only on the target descriptions.

Evaluation Metrics We use a set of automatic evaluation metrics to assess the quality of generated character descriptions following previous work (Papoudakis et al., 2024).

PRISMA (Mahon and Lapata, 2024) is an LLM-as-a-judge metric that evaluates factual accuracy. It

²BookWorm has two tasks (description and analysis); we use the description partition, which contains more examples.

³<https://github.com/booknlp/booknlp>

extracts facts from the generated description and assesses their correctness against the gold standard, then extracts facts from the gold standard and checks whether the generated output supports them. These precision and recall scores are combined into PRISMA F1. We use GPT-4o-mini for fact extraction and Bespoke-MiniCheck-7B (Tang et al., 2024) for fact validation, which achieves state-of-the-art performance on the LLM-AggreFact benchmark (Tang et al., 2024).

We also evaluate factual precision against the input story using an entailment-based NLI (natural language inference) metric. For each extracted fact, we calculate entailment scores against story chunks, taking the maximum score across all chunks. A fact is considered grounded if its score exceeds 0.5. We use Bespoke-MiniCheck-7B with 1,024-token chunks for this evaluation. We complement NLI metrics with **QA-based evaluation** (Deutsch et al., 2021) which measures whether the generated description contains the key information needed to answer questions derived from the reference. We use GPT-4o-mini to generate QA-pairs from the reference and DeBERTaV3 (He et al., 2023) fine-tuned on SQuADv2 (Rajpurkar et al., 2018) for question-answering.

We further report the **entity-mention** F1 score as a measure of entity-level coverage. Specifically, we compute precision as the proportion of the entities mentioned in the generated description that also appear in the gold-standard description, and recall as the proportion of gold-standard entities recovered by the generated output; these then combine into entity-mention F1. We also report **Rouge-L** (Lin, 2004), which measures the longest common subsequence with the reference descriptions.

To evaluate the *quality of the QA-guided trace*, we compare QA-pairs produced by the reasoning model against QA-pairs derived from reference descriptions. We measure whether each generated pair is supported by the gold QA set (precision), and how many gold QA pairs are covered by the generated set (recall). We use GPT-4o-mini in an LLM-as-a-judge setting for this evaluation. Our prompts and further details are in Appendix A.

5 Results

Table 2 reports performance on BookWorm across retrieval, hierarchical, incremental, and long-context settings, with and without reasoning.

Built-in reasoning hurts faithfulness. Across settings, enabling the model’s *built-in* reasoning consistently reduces overall quality compared to an empty trace. For example, under Coref-32k, built-in reasoning lowers Rouge-L (18.44 → 16.54) and NLI (52.58 → 49.94). A similar pattern holds for Hierarchical-16k, where built-in reasoning decreases QA (15.79 → 15.16) and Rouge-L (18.22 → 17.33). The effect is also present in the long-context setting for Lead-128k, with NLI dropping from 69.29 to 67.17 and Rouge-L from 18.11 to 16.43. These results suggest that default “thinking” traces introduce verbosity or unsupported details, harming grounding in long narratives (see Table 10 in Appendix B for discussion on reasoning traces).

QA-guided reasoning improves grounding and coverage. In contrast, our guided-QA traces improve evidence-sensitive metrics, particularly QA F1 and entity coverage. For BM25-32k, guided-QA yields substantial gains in EntMent (34.06 → 36.23) and QA (13.56 → 14.66), while also improving NLI (58.58 → 59.96), indicating better grounding in the retrieved evidence. For Coref-32k, guided-QA improves QA (13.96 → 15.01) while maintaining comparable EntMent. For hierarchical processing, guided-QA provides gains in EntMent (36.56 → 37.67), maintains comparable QA F1 and PRISMA, demonstrating that reasoning helps even when the model already aggregates chunk-level summaries. In the long-context setting, guided-QA with GRPO-trained traces improves over the baseline across all metrics except R-L (PRISMA: 17.02 → 19.30, QA: 13.68 → 16.22). Overall, guided-QA is most beneficial when the context selection step surfaces relevant evidence, but the generator needs help integrating it into a coherent, grounded description.

Trace quality and generator training affect different metrics. Comparing zero-shot and GRPO-trained traces reveals that better traces translate into better descriptions, though gains are metric-dependent. Under BM25-32k, GRPO traces improve PRISMA (17.09 → 18.38) while preserving high EntMent, whereas SFT on descriptions primarily benefits surface-overlap metrics (e.g., Rouge-L: 18.18 → 18.99) with weaker effects on factual scores. A similar pattern emerges for Coref-32k, where SFT achieves the highest Rouge-L (19.26) but GRPO-trained traces improve QA (13.96 → 14.86) and yield the best EntMent (37.66). This suggests that trace-level optimization and generation-

	Method	Trace	Desc	PRISMA	QA	NLI	EntMent	Rouge-L
Qwen-3-8b	No Context	—	ZS	11.97	10.31	—	25.97	17.55
	Lead-32k	—	ZS	15.72	12.35	61.01	30.60	17.79
	BM25-32k	—	ZS	17.09	13.56	58.58	34.06	18.18
	BM25-32k	—	SFT	16.17 [†]	13.26	60.00 [†]	33.11 [†]	18.99 [†]
	+ reasoning	built-in	ZS	17.06	13.50	58.44	35.13	16.51 [†]
	+ guided-QA	ZS	ZS	17.84	14.66 [†]	59.96 [†]	36.23 [†]	17.43 [†]
	+ guided-QA	GRPO	ZS	18.38 [†]	15.18 [†]	59.90 [†]	37.05 [†]	17.59 [†]
	+ guided-QA	GRPO	SFT	18.27 [†]	15.15 [†]	60.46 [†]	36.60 [†]	17.87 [†]
	Coref-32k	—	ZS	19.11	13.96	52.58	36.12	18.44
	Coref-32k	—	SFT	17.46 [†]	14.36	52.54	35.59	19.26[†]
	+ reasoning	built-in	ZS	18.59 [†]	13.94	49.94 [†]	35.88	16.54 [†]
	+ guided-QA	ZS	ZS	18.63 [†]	15.01 [†]	50.60 [†]	36.95	17.50 [†]
	+ guided-QA	GRPO	ZS	19.49	14.86 [†]	51.23 [†]	37.66 [†]	18.01 [†]
	+ guided-QA	GRPO	SFT	19.32	15.10 [†]	52.00	37.58 [†]	18.42
	Hierarchical-16k	—	ZS	19.99	15.79	70.57	36.56	18.22
	+ reasoning	built-in	ZS	19.74	15.16 [†]	71.15	35.82	17.33 [†]
	+ guided-QA	ZS	ZS	19.82	16.20	70.74	37.67 [†]	17.90 [†]
	+ guided-QA	GRPO	ZS	18.94 [†]	15.36	67.92 [†]	36.49	18.19
	Incremental-16k	—	ZS	17.64	14.42	69.68	35.59	17.58
	+ reasoning	built-in	ZS	16.03 [†]	13.36 [†]	66.46 [†]	33.37 [†]	15.76 [†]
+ guided-QA	ZS	ZS	16.80 [†]	14.56	68.20 [†]	34.92	16.98 [†]	
+ guided-QA	GRPO	ZS	16.59 [†]	13.63 [†]	66.05 [†]	34.51 [†]	17.42	
Lead-128k (w/ YaRN)	—	ZS	17.02	13.68	69.29	34.94	18.11	
+ reasoning	built-in	ZS	16.93	13.65	67.17 [†]	34.20	16.43 [†]	
+ guided-QA	ZS	ZS	19.32 [†]	15.77 [†]	71.55 [†]	37.80[†]	17.77 [†]	
+ guided-QA	GRPO	ZS	19.30 [†]	16.22 [†]	70.51 [†]	37.12 [†]	18.39 [†]	
GPT-4.1 mini	No context	—	ZS	17.36	12.49	—	29.35	17.41
	Full context	—	ZS	22.05	16.67	75.53	35.41	17.71

Table 2: Results on BookWorm. Trace: — (none), built-in (model’s default), ZS (zero-shot guided-QA), GRPO (GRPO-trained guided-QA). Desc: ZS (zero-shot) or SFT (supervised fine-tuning). [†] indicates statistically significant difference from the corresponding baseline without reasoning (i.e., first row for each method group); **bold** indicates best per metric.

level training target complementary aspects of output quality.

Benefits in incremental settings are limited. Incremental processing presents a challenging setting where guided-QA shows limited benefit. Both built-in and guided reasoning degrade PRISMA compared to the baseline (17.64 \rightarrow 16.03 and 16.80, respectively). We hypothesize that the sequential updating mechanism struggles to incorporate reasoning traces effectively, as each update step must reconcile new QA pairs with a partial description. This suggests that our approach is best suited for settings where evidence can be aggregated before generation rather than integrated incrementally.

Entity coverage improves even against proprietary models. For reference, we include GPT-4.1-mini with full context access, which achieves the highest PRISMA (22.05), QA F1 (16.67) and NLI (75.53). However, guided-QA with the smaller

Qwen-3-8B model narrows this gap, achieving the best EntMent score overall (37.80 vs. 35.41). This indicates that structured reasoning traces improve entity coverage even compared to larger proprietary models with longer context windows.

Faithful QA traces improve descriptions across QA strategies. Table 4 compares alternative question-generation strategies for the reasoning model and relates trace quality to downstream description performance (the generator is kept zero-shot in all settings). We also report an *Oracle* upper bound, where QA pairs are extracted from the gold descriptions and used directly as traces, highlighting the remaining headroom when perfect intermediate signals are available.

Overall, optimizing the reasoner improves both trace quality and description quality. In particular, GRPO yields the best trace recall and the highest trace F1 (16.96), outperforming both zero-shot guided reasoning and SFT on traces. This

Model	Method	Trace	Desc	PRISMA	QA	NLI	EntMent	Rouge-L
Qwen-3-8b	No Context	—	ZS	7.69	2.92	—	19.83	16.40
	Coref-32k	—	ZS	23.13	13.03	50.01	35.90	17.44
	+ reasoning	built-in	ZS	22.26 [†]	12.52	46.84 [†]	33.50 [†]	16.02 [†]
	+ guided-QA	ZS	ZS	21.88 [†]	13.11	49.63	35.46	17.24 [†]
	+ guided-QA	GRPO	ZS	22.63 [†]	13.08 [†]	49.70	35.38 [†]	17.33 [†]
	Lead-128k (w/ YaRN)	—	ZS	22.22	13.70	66.39	35.48	17.21
	+ reasoning	built-in	ZS	18.08 [†]	10.97 [†]	66.98	31.18 [†]	15.75 [†]
	+ guided-QA	ZS	ZS	23.72 [†]	14.62 [†]	68.57 [†]	36.84[†]	17.47 [†]
	+ guided-QA	GRPO	ZS	23.78 [†]	14.42 [†]	68.09 [†]	36.18 [†]	17.49[†]
GPT-4.1-mini	No context	—	ZS	10.01	3.98	—	21.46	16.10
	Full context	—	ZS	28.49	16.08	72.67	28.49	16.80

Table 3: Results on CroSS. Trace: — (none), built-in (model’s default), ZS (zero-shot guided-QA), GRPO (GRPO-trained guided-QA). Desc: ZS (zero-shot) or SFT (supervised fine-tuning). [†]indicates statistically significant difference from the corresponding baseline without reasoning (i.e., first row for each method group); **bold** indicates best per metric.

Method	QA reasoning				Description	
	# QA	Prec	Rec	F1	PRISMA	QA
No reasoning	—	—	—	—	15.20	14.42
No chunking	6.50	16.10	16.93	16.50	14.02	14.03
guided-QA	8.60	15.07	17.68	16.27	14.45	14.45
guided-QA (SFT)	6.60	14.84	15.98	15.38	14.11	14.08
guided-QA (GRPO)	11.07	15.25	19.10	16.96	15.25	14.79
Oracle	7.51	—	—	—	44.81	40.61

Table 4: Comparison of question generation methods for the reasoning model on the BookWorm dataset (validation set) with coreference-based retrieval. The generation model is zero-shot in all experiments.

improvement translates downstream: conditioning on GRPO traces produces the best QA F1 (14.79) and PRISMA score (15.25), though the PRISMA gains over no reasoning are only marginal (15.20 \rightarrow 15.25). By contrast, increasing the number of QA pairs alone is not sufficient: guided-QA generates more questions than the no chunking approach but has lower precision and total F1 score. These results suggest that downstream gains are driven by *faithful* and *informative* traces rather than by trace length (see question generation experiments in Appendix B). Finally, the oracle traces dramatically outperform all automatic traces, indicating substantial room for improvement in trace generation and verification.

Guided reasoning improves transfer to CroSS. Table 3 evaluates whether models tuned on BookWorm transfer to CroSS (Yuan et al., 2024). Using Qwen-3-8B, character-focused context strategies yield substantial gains over no context (e.g., Coref-32k PRISMA: 7.69 \rightarrow 23.13). Consistent with BookWorm, built-in reasoning degrades grounding metrics (Coref-32k NLI: 50.01 \rightarrow 46.84; Lead-128k EntMent: 35.48 \rightarrow 31.18), while guided-QA im-

proves or keeps comparable performance across settings. For Lead-128k, guided-QA with GRPO traces achieves the best PRISMA (23.78) and QA F1 (14.42) among Qwen-3-8B configurations, demonstrating that our approach benefits long-context processing on unseen data. GPT-4.1-mini with full context provides a strong upper bound (PRISMA 28.49, NLI 72.67), though Qwen-3-8B with guided reasoning achieves competitive or superior entity coverage (EntMent: 36.84 vs. 28.49), confirming that structured traces improve grounding even against larger models.

6 Conclusion

We address the challenge of generating accurate character descriptions from book-length narratives by proposing a modular framework that decouples reasoning from generation. Our experiments reveal that enabling the built-in reasoning mode of current LLMs often degrades performance on character description generation, with empty reasoning traces yielding more faithful outputs across multiple long-context settings. To address this, we introduce QA-guided reasoning, a two-stage approach consisting of (1) a reasoning model that generates structured question-answer traces capturing salient character information, and (2) a generation model that conditions on these traces to produce final descriptions.

A key advantage of our approach is its training strategy: the reasoning model is optimized directly with Group Relative Policy Optimization on gold-standard character descriptions and automatically derived QA traces, eliminating the need to define rewards over the final generated text, which is a particularly challenging problem for open-ended

generation tasks. The generation model can be trained with supervised fine-tuning by injecting reasoning traces between thinking tokens. Experiments on two datasets (BookWorm and CroSS) demonstrate that QA-guided reasoning improves faithfulness and informativeness across multiple evaluation metrics. Our analysis shows that trace quality directly impacts downstream description quality, with GRPO-optimized traces yielding the best performance. The approach also transfers effectively to out-of-distribution data, suggesting that the learned reasoning patterns generalize beyond the training distribution.

In future work we would like to explore alternative reasoning structures beyond QA pairs, investigate other reinforcement learning algorithms and reward formulations, and develop character-specific evaluation metrics that better capture narrative understanding.

Limitations

Our evaluation relies on automatic metrics following established practices in prior work. We employ question-answering and fact-based metrics (PRISMA) to assess descriptions against gold standards, entailment-based metrics (NLI) to measure grounding in the input story, and standard surface-level metrics such as entity-mention F1 and Rouge-L. While these metrics provide useful signals, they have important limitations for evaluating long-context generation. Human evaluation, though more reliable, remains prohibitively expensive and difficult to scale to the thousands of examples required for robust assessment. Automatic metrics, conversely, struggle to capture nuanced aspects of character understanding, such as narrative coherence, implicit trait inference, and the integration of evidence across distant spans. Future work should develop character-specific evaluation frameworks that better capture narrative understanding and can be applied at scale.

Our training approach uses GRPO to optimize the reasoning model. While GRPO is computationally efficient and performs well in our experiments, other reinforcement learning algorithms (e.g., PPO, DPO) or alternative reward formulations may yield further improvements. We leave exploration of these alternatives to future work.

Additionally, our approach requires silver-standard QA traces derived from gold descriptions during training. These gold descriptions are global,

for the entire story, and we do not have annotations aligning them with specific passages of the story. Future work should explore aligning descriptions or the corresponding QA-pairs with input passages and restricting GRPO training to passage-aligned QA-pairs. Finally, in settings where high-quality character descriptions are unavailable, alternative supervision strategies (e.g., weak supervision from plot summaries or character wikis) may be necessary. Investigating such strategies would broaden the applicability of our framework.

Acknowledgments

This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) and the University of Edinburgh, School of Informatics and School of Philosophy, Psychology & Language Sciences. Computing resources were provided by the Edinburgh International Data Facility (EIDF) and the Data-Driven Innovation Programme at the University of Edinburgh. Access to EIDF was facilitated through the University of Edinburgh’s Generative AI Laboratory GAIL Fellow scheme. We thank the metareviewer and anonymous reviewers for their valuable feedback and comments. Lapata gratefully acknowledges the support of the UK Engineering and Physical Sciences Research Council (grant EP/W002876/1).

References

- David Bamman, Olivia Lewke, and Anya Mansoor. 2020. [An annotated dataset of coreference in English literature](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 44–54, Marseille, France. European Language Resources Association.
- David Bamman, Brendan O’Connor, and Noah A. Smith. 2013. [Learning latent personas of film characters](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361, Sofia, Bulgaria. Association for Computational Linguistics.
- Sabyasachee Baruah and Shrikanth Narayanan. 2023. [Character coreference resolution in movie screenplays](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10300–10313, Toronto, Canada. Association for Computational Linguistics.
- Sabyasachee Baruah and Shrikanth Narayanan. 2025. [CHATTER: A character-attribution dataset for narrative understanding](#). In *Proceedings of the The 7th*

- Workshop on Narrative Understanding*, pages 52–63, Albuquerque, New Mexico. Association for Computational Linguistics.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Faeze Brahman, Meng Huang, Oyvind Tafjord, Chao Zhao, Mrinmaya Sachan, and Snigdha Chaturvedi. 2021. “let your characters tell their story”: A dataset for character-centric narrative understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1734–1752, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yapei Chang, Yekyung Kim, Michael Krumdick, Amir Zadeh, Chuan Li, Chris Tanner, and Mohit Iyer. 2025. Bleuberi: Bleu is a surprisingly effective reward for instruction following. *arXiv preprint arXiv:2505.11080*.
- Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyer. 2023. Boookscore: A systematic exploration of book-length summarization in the era of llms. In *International Conference on Learning Representations*.
- Snigdha Chaturvedi, Shashank Srivastava, Hal Daume III, and Chris Dyer. 2015. Modeling dynamic relationships between characters in literary novels. *arXiv preprint arXiv:1511.09376*.
- Nuo Chen, Yan Wang, Haiyun Jiang, Deng Cai, Yuhan Li, Ziyang Chen, Longyue Wang, and Jia Li. 2023. Large language models meet harry potter: A dataset for aligning dialogue agents with characters. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8506–8520, Singapore. Association for Computational Linguistics.
- Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359.
- Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2021. Towards question-answering as an automatic metric for evaluating the content quality of a summary. *Transactions of the Association for Computational Linguistics*, 9:774–789.
- Yiran Ding, Li Lina Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending llm context window beyond 2 million tokens. *Preprint*, arXiv:2402.13753.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.
- Yao Fu. 2024. Challenges in deploying long-context transformers: A theoretical peak performance analysis. *arXiv preprint arXiv:2405.08944*.
- Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2025. How to train long-context language models (effectively). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7376–7399, Vienna, Austria. Association for Computational Linguistics.
- Alexander Gurung and Mirella Lapata. 2024. CHIRON: Rich character representations in long-form narratives. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8523–8547, Miami, Florida, USA. Association for Computational Linguistics.
- Alexander Gurung and Mirella Lapata. 2025. Learning to reason for long-form story generation. *arXiv preprint arXiv:2503.22828*.
- Daniel Han, Michael Han, and Unsloth Team. 2023. Unsloth.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Jian Hu, Xibin Wu, Zilin Zhu, Weixun Wang, Dehao Zhang, Yu Cao, and 1 others. 2024. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*.
- Fantine Huot, Joshua Maynez, Shashi Narayan, Reinald Kim Amplayo, Kuzman Ganchev, Annie Priyadarshini Louis, Anders Sandholm, Dipanjan Das, and Mirella Lapata. 2023. Text-blueprint: An interactive platform for plan-based conditional generation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 105–116, Dubrovnik, Croatia. Association for Computational Linguistics.
- Naoya Inoue, Charuta Pethe, Allen Kim, and Steven Skiena. 2022. Learning and evaluating character representations in novels. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1008–1019, Dublin, Ireland. Association for Computational Linguistics.
- Mohit Iyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former Friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the 2016 Conference of the North American Chapter*

- of the Association for Computational Linguistics: Human Language Technologies, pages 1534–1544, San Diego, California. Association for Computational Linguistics.
- Evgeny Kim and Roman Klinger. 2019. [Frowning Frodo, wincing Leia, and a seriously great friendship: Learning to classify emotional relationships of fictional characters](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 647–653, Minneapolis, Minnesota. Association for Computational Linguistics.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.
- Vincent Labatut and Xavier Bost. 2019. [Extraction and analysis of fictional character networks: A survey](#). *ACM Comput. Surv.*, 52(5).
- Kunze Li and Yu Zhang. 2024. [Planning first, question second: An LLM-guided method for controllable question generation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4715–4729, Bangkok, Thailand. Association for Computational Linguistics.
- Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhui Chen. 2024. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Danyang Liu, Juntao Li, Meng-Hsuan Yu, Ziming Huang, Gongshen Liu, Dongyan Zhao, and Rui Yan. 2020. A character-centric neural model for automated story generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1725–1732.
- Dongqi Liu, Xi Yu, Vera Demberg, and Mirella Lapata. 2025. [Explanatory summarization with discourse-driven planning](#). *Transactions of the Association for Computational Linguistics*, 13:1146–1170.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Louis Mahon and Mirella Lapata. 2024. [A modular approach for multimodal summarization of TV shows](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8272–8291, Bangkok, Thailand. Association for Computational Linguistics.
- Giuliano Martinelli, Tommaso Bonomo, Pere-Lluís Huguet Cabot, and Roberto Navigli. 2025. [BOOK-COREF: Coreference resolution at book scale](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24526–24544, Vienna, Austria. Association for Computational Linguistics.
- Shashi Narayan, Joshua Maynez, Reinald Kim Amplayo, Kuzman Ganchev, Annie Louis, Fantine Huot, Anders Sandholm, Dipanjan Das, and Mirella Lapata. 2023. [Conditional generation with a question-answering blueprint](#). *Transactions of the Association for Computational Linguistics*, 11:974–996.
- Kimia Noorbakhsh, Joseph Chandler, Pantea Karimi, Mohammad Alizadeh, and Hari Balakrishnan. 2025. Savaal: Scalable concept-driven question generation to enhance human learning. *arXiv preprint arXiv:2502.12477*.
- Eric W Noreen. 1989. Computer intensive methods for hypothesis testing: An introduction. Wiley, New York, 19:21.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Argyrios Papoudakis, Mirella Lapata, and Frank Keller. 2024. [BookWorm: A dataset for character description and analysis](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4471–4500, Miami, Florida, USA. Association for Computational Linguistics.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. [Know what you don’t know: Unanswerable questions for SQuAD](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.
- Yisi Sang, Xiangyang Mou, Mo Yu, Dakuo Wang, Jing Li, and Jeffrey Stanton. 2022. [MBTI personality prediction for fictional characters using movie scripts](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6715–6724, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

- Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. [Character-LLM: A trainable agent for role-playing](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13153–13187, Singapore. Association for Computational Linguistics.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Parshin Shojaee, Aneesh Jain, Sindhu Tipirneni, and Chandan K Reddy. 2023. Execution-based code generation using deep reinforcement learning. *arXiv preprint arXiv:2301.13816*.
- Marcin Skowron, Martin Trapp, Sabine Payr, and Robert Trapp. 2016. Automatic identification of character types from film dialogs. *Applied Artificial Intelligence*, 30(10):942–973.
- Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. [VeriScore: Evaluating the factuality of verifiable claims in long-form text generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9447–9474, Miami, Florida, USA. Association for Computational Linguistics.
- Liyan Tang, Philippe Laban, and Greg Durrett. 2024. [MiniCheck: Efficient fact-checking of LLMs on grounding documents](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847, Miami, Florida, USA. Association for Computational Linguistics.
- Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.
- Runchu Tian, Yanghao Li, Yuepeng Fu, Siyang Deng, Qinyu Luo, Cheng Qian, Shuo Wang, Xin Cong, Zhong Zhang, Yesai Wu, Yankai Lin, Huadong Wang, and Xiaojiang Liu. 2025. [Distance between relevant information pieces causes bias in long-context LLMs](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 521–533, Vienna, Austria. Association for Computational Linguistics.
- Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.
- Krishnapriya Vishnubhotla, Adam Hammond, Graeme Hirst, and Saif Mohammad. 2024. [The emotion dynamics of literary novels](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2557–2574, Bangkok, Thailand. Association for Computational Linguistics.
- Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*.
- Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.
- Mo Yu, Jiangnan Li, Shunyu Yao, Wenjie Pang, Xiaochen Zhou, Zhou Xiao, Fandong Meng, and Jie Zhou. 2023. [Personality understanding of fictional characters during book reading](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14784–14802, Toronto, Canada. Association for Computational Linguistics.
- Xinfeng Yuan, Siyu Yuan, Yuhan Cui, Tianhe Lin, Xintao Wang, Rui Xu, Jiangjie Chen, and Deqing Yang. 2024. [Evaluating character understanding of large language models via character profiling from fictional works](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8015–8036, Miami, Florida, USA. Association for Computational Linguistics.
- Weiwei Zhang, Jackie Chi Kit Cheung, and Joel Oren. 2019. Generating character descriptions for automatic summarization of fiction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7476–7483.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. 2025. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*.

Lixing Zhu, Runcong Zhao, Lin Gui, and Yulan He. 2023. [Are NLP models good at tracing thoughts: An overview of narrative understanding](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10098–10121, Singapore. Association for Computational Linguistics.

A Implementation Details

Training and Inference We use the OpenRLHF (Hu et al., 2024) library for GRPO training with the hyperparameters listed in Table 5. We also employ the Qwen-3-8b model to provide LLM-as-a-judge rewards during GRPO-training. For supervised finetuning, we use LoRA (Hu et al., 2022) from the unsloth (Han et al., 2023) library. We report the supervised finetuning hyperparameters in Table 6. For inference, we use the vLLM engine (Kwon et al., 2023) with sample decoding and temperature 0.4. We evaluate the statistical significance of the results using two-sided paired approximate randomization test (10,000 permutations and $\alpha = 0.05$) (Noreen, 1989), based on the mean value across 4 inference seeds. We used four H200 Nvidia GPUs for GPRO training and a single H100 or H200 GPU for all the inference experiments and SFT training.

Hyperparameter	Value
actor learning rate	5×10^{-7}
KL coefficient	0.01
train batch size	64
samples per prompt	8
prompt max length	17,684
generation max length	2,048

Table 5: Training hyperparameters for Group Relative Policy Optimization (GRPO) training.

Hyperparameter	Value
learning rate	10^{-6}
max input length	32,768
batch size	1
gradient accumulation steps	8
alpha	256
rank	128
lora dropout	0.1

Table 6: Training hyperparameters for supervised finetuning (SFT) training with LoRA.

Prompts We provide the prompts used for our experiments in Tables 7 and 8.

Evaluation We use the VeriScore (Song et al., 2024) codebase to extract facts (prompts are adjusted based on Papoudakis et al. (2024)) for both PRISMA and NLI metrics. Fact verification is performed using the MiniCheck (Tang et al., 2024) model in both metrics. We use rouge-score⁴ implementation for Rouge-L metric and NLTK⁵ toolkit

⁴<https://github.com/google-research/google-research/tree/master/rouge>

⁵<https://www.nltk.org/>

Context: {context}

Describe character {character} from book {book} based on the given context.

Return your output as a single paragraph (close to {length} words) including the important information.

Table 7: Character description generation prompt template. We replace the variable length with the average output length of the corresponding dataset.

Context: {context}

Your task is to generate question-answer pairs about character: {character} from book: {book} given the above chunk of the book. You should focus on understanding aspects of the character (e.g., role, relationships, personality, events) that are mentioned in the context. Each qa-pair should be labelled as role, relationship, personality, event or other. We provide the definitions for these below.

Definitions:

Role: defines what part the character plays in the story, narrator, major/minor character.

Relationship: connections the character has with others, such as friendships or family ties.

Personality: character’s behavior, traits, and attributes.

Event: actions and decisions the character is involved in throughout the story.

Other: any other fact that doesn’t belong to the above categories.

Output format:

Q1: <question> E1: <explanation> A1: <answer> T1: <type>

Q2: <question> E2: <explanation> A2: <answer> T2: <type>

...

Generate an explanation, 1-2 sentences that fully justify your answer, do not simply repeat the answer. Type of qa has to be one of Role, Relationship, Personality, Event or Other. The answer should be short 1-4 words. Generate QA-pairs only related to character: {character}. Generate QA-pairs only for information mentioned in the provided context. Do not include unanswered questions.

The questions must mention the character’s name: {character}. Do not generate repetitive QA-pairs with same answer. If the character is not mentioned, simply return None.

Table 8: QA generation prompt template. We adopt the definitions for the different categories from Papoudakis et al. (2024)

for named-entity extraction in entity-mention F1 calculation.

B Additional Experiments and Statistics

QA Reasoning Ablation We run an ablation study to demonstrate the effect of the different components in the reasoning trace. We use the Qwen-3-

Method	QA Reasoning				Description		
	#QA	P	R	F1	PRI	QA	Ent
guided-QA	8.60	15.07	17.68	16.27	14.45	14.45	30.41
w/o expl.	12.05	13.89	16.93	15.26	14.34	15.01	29.94
w/o types	11.44	12.46	15.87	13.95	13.41	14.34	30.37
w/o expl., types	15.46	13.90	12.16	13.61	13.71	14.92	30.08
w/o expl., ans., types	13.44	—	—	—	14.02	14.22	29.62

Table 9: Ablation of QA reasoning components using Qwen-3-8b with coreference retrieval on BookWorm validation set (zero-shot). P/R/F1 = reasoning precision/recall/F1; PRI = PRISMA F1; Ent = EntMent F1.

8B model as a backbone with a coreference-based retrieval method to extract context and employ the guided reasoning method in zero-shot mode.

Table 9 examines the contribution of each component in the QA reasoning trace: explanations, question types, and answers. The full guided-QA method achieves the best PRISMA (14.45) and EntMent (30.41), indicating that all components contribute to faithful and entity-rich descriptions. Removing explanations (w/o expl.) increases the number of generated QA pairs (8.60 \rightarrow 12.05) and improves QA F1 (14.45 \rightarrow 15.01), but reduces precision (15.07 \rightarrow 13.89) and recall (17.68 \rightarrow 16.93), suggesting that explanations help maintain informative and precise reasoning traces.

Removing question types (w/o types) causes a drop in reasoning F1 (16.27 \rightarrow 13.95) and PRISMA (14.45 \rightarrow 13.41), indicating that typed questions encourage topical diversity and more informative traces. Removing both explanations and types (w/o expl., types) generates the most QA pairs (15.46) but with lower precision (13.90) and recall (12.16), yielding mixed downstream results.

Finally, retaining only questions (w/o expl., ans., types) prevents reasoning evaluation (no answers to verify) and degrades PRISMA (14.02), confirming that answers are essential for grounding the generation model. Overall, each component serves a distinct role: types encourage diversity, explanations improve precision, and answers provide the factual content that guides description generation.

Reasoning Statistics Table 10 reports summary statistics for intermediate reasoning traces and the final character descriptions when conditioning on the retrieved context in BookWorm. Overall, the *description* outputs remain similar in length across settings (approximately 109–118 tokens) and exhibit comparable lexical diversity (about 71–73% unique unigrams) for the no-reasoning baseline and both guided-QA variants. This suggests that per-

Method	Reasoning			Description	
	#QA	Tok	Uni	Tok	Uni
None	—	—	—	111	71
built-in	—	270	60	109	78
guided-QA (ZS)	8	391	36	118	73
guided-QA (GRPO)	11	535	31	111	71

Table 10: Statistics for generated traces and their corresponding descriptions for different reasoning methods using retrieved context on BookWorm validation set. #QA refers to the number of QA pairs; Tok is the average number of tokens and Uni the proportion of unique unigrams.

formance differences in our main experiments are unlikely to be driven by longer or more lexically diverse descriptions, but instead by the content and structure of the intermediate reasoning signal.

In contrast, the *reasoning* traces differ substantially. Built-in reasoning produces moderately long traces (270 tokens) with higher unigram diversity than guided-QA, whereas guided-QA traces are much longer (391–535 tokens) but show markedly lower unigram diversity (31–36%), which is largely attributable to the structured QA format (e.g., repeated question templates and answer markers). Comparing the two guided-QA variants, GRPO yields longer traces (11 QA pairs; 535 tokens) than the zero-shot reasoner (8 QA pairs; 391 tokens), while maintaining similar diversity, which suggests that optimization encourages the model to select more, higher-yield QA pairs but without increasing verbosity. Finally, built-in reasoning leads to the highest description unigram diversity (78%) despite lower faithfulness in our main results, consistent with the interpretation that unstructured deliberation introduces additional (and potentially unsupported) details rather than improving grounding.

Experiments with Gemma-3 We further evaluate our approach using the Gemma-3-12b-it (Team et al., 2025) model for coreference-based retrieval and Lead-128k methods on the BookWorm dataset. We report a reasoning experiment in which the model is prompted to explicitly reason, generating a chain-of-thought inside `<think> ... </think>` tokens before producing the character description. Although the Gemma-3 model is not trained to explicitly reason, we found that it consistently follows the reasoning format in all the experiments. The results in Table 11 demonstrate that explicit reasoning harms performance, for exam-

Method	PRISMA	QA	EntMent	R-L
Coref-32k	13.65	12.49	27.93	16.20
+ reasoning	11.90	11.11	25.60	15.49
+ guided-QA	13.10	13.60	28.82	16.61
Lead-128k	11.76	11.78	26.39	15.99
+ reasoning	11.16	10.67	25.66	15.16
+ guided-QA	13.12	13.32	29.34	16.62

Table 11: Results on BookWorm (validation set) using Gemma-12b-it model. Zero-shot experiments without any reasoning trace, explicitly prompting the model to reason and QA-guided reasoning for coref-based retrieval and Lead-128k methods.

ple, both PRISMA (11.76 \rightarrow 11.16) and QA-F1 (11.78 \rightarrow 10.67) degrade for Lead-128k. In contrast, zero-shot QA-guided reasoning in the coreference method improves performance over the corresponding approaches in all metrics except PRISMA, achieving also the highest QA-F1 score (13.60). Similarly, QA-guided reasoning for Lead-128k improves over the corresponding baselines across all metrics, achieving the highest EntMent (29.34) and R-L (16.62). Overall, these results indicate that findings from the main experiments also apply to a model from a different family, which is not trained for explicit reasoning, and that the proposed QA-guided method can also be effective under this setting.

QA Generation Methods We compare different question generation methods, evaluating the corresponding reasoning traces. Specifically, we report experiments with **Savaal** (Noorbakhsh et al., 2025), a multi-step topic-driven approach for QA generation in long documents. This approach first extracts topics from each chunk of the input and then uses each topic to retrieve relevant passages. The retrieved passages and the corresponding topic are finally used to generate QA-pairs.

Next, we evaluate a plan-based QA generation method based on **PFQS** (planning first, question second) (Li and Zhang, 2024). We prompt a model to extract sentences from the given input that serve as candidate evidence for answers. The extracted sentences are provided together with the input context to generate QA pairs. We evaluate both approaches using the same output format, adding an explanation and a type to each QA-pair (see Table 8). We also split the input context into chunks of 16k tokens.

Table 12 shows the evaluation of the different methods and their average number of questions.

Method	# QA	Precision	Recall	F1
guided-QA	8.60	15.07	17.68	16.27
Savaal	12.01	13.92	15.23	14.54
Plan-based QA	11.90	11.71	17.34	13.98

Table 12: Experiments with different QA generation methods using coref-based retrieval on BookWorm (validation set). We report the average number of QA-pairs along with precision, recall and F1 score for reasoning evaluation.

Method	# QA	PRISMA	QA	EntMent	R-L
guided-QA (fixed # QA)	6.40	13.34	13.58	30.11	17.10
	13.50	13.46	14.03	30.53	17.05
	20.20	13.49	14.07	29.63	16.98
	26.20	13.56	14.45	30.27	17.14
	32.00	13.67	14.44	29.78	17.06
	35.10	13.71	14.08	29.99	17.01
	38.00	13.41	14.31	29.99	17.10

Table 13: QA-guided reasoning with different number of generated QA-pairs using BM25-32k as input context on BookWorm (validation set). We report the average number of QA-pairs along with PRISMA, QA-F1, EntMent and R-L for character description evaluation.

The guided-QA method yields the highest precision, recall, and F1 (16.27) despite having the lowest number of QA pairs. The Savaal method obtains a moderate F1 score but performs worse than guided-QA, while plan-based generation has competitive recall, but precision is significantly lower (11.71). Overall, these results demonstrate that simple chunk-based question generation can outperform topic- and plan-based methods, yielding more precise and informative QA-pairs even with fewer generated questions.

Number of Generated QA Pairs We evaluate how the number of QA pairs in the reasoning trace affects downstream character description performance. We adjust the question-generation prompt defined in Table 8 to explicitly instruct the model to generate a fixed number of QA pairs. Note that the number of generated questions does not always match the requested number, as there are chunks where the character is not mentioned or the model fails to generate the correct number of questions. We use BM25-32k as input context, and both QA reasoning and generation models are evaluated in a zero-shot setting.

Results are shown in Table 13. We find that different metrics follow different patterns: PRISMA increases gradually, peaks at 35 questions and then drops at 38 questions. QA and R-L both have the

Method	Trace	Descr	PRISMA	QA	EntMent
None	–	ZS	13.63	13.29	27.54
guided-QA	ZS	ZS	13.67	13.82	29.87
guided-QA	GRPO	ZS	13.71	14.24	29.32
guided-QA	GRPO	GRPO	14.22	14.50	29.82

Table 14: Results on BookWorm (validation set) using BM25-32k as input context. We report reasoning and description generation method along with PRISMA, QA-F1 and EntMent for each experiment.

highest value at approximately 26 questions, after which they decrease, though R-L has only marginal differences across experiments. EntMent peaks at 13 questions, and fluctuates as the number of questions increases. Overall, there is no single pattern consistent across metrics, however most metrics achieve best performance in the range of 13 – 26 questions, except for PRISMA, which continues to improve until a larger number of questions.

GRPO for Description Generation Our approach is motivated by the observed decline in the performance of built-in reasoning in the character description task. Thus, we focus on the reasoning model that guides the generation process, while the generation model is employed in either a zero-shot setting or with supervised fine-tuning (SFT). To extend our analysis, we experiment with GRPO training for the generation model, similar to the training setup of the reasoning model. Specifically, we use the F1 score as a reward signal, extracting QA-pairs from the generated description and verifying against the gold-standard and vice versa. We use BM25-32k as the input context for these experiments.

Results are shown in Table 14. We observe that GRPO-training of the description generation model leads to increased scores in PRISMA and QA, and a marginally lower EntMent score, indicating that GRPO is also effective for the generation model, improving factual scores. This is consistent with our analysis in the results section, where we observed that SFT and GRPO training target different aspects of the output: GRPO primarily improves factual metrics (e.g., PRISMA) while SFT benefits surface-level metrics (e.g., R-L).

C Examples

We show examples of generated outputs and reasoning traces in Tables 15, 16, 17 and 18.

Character: Demon Copperhead **Book:** Demon Copperhead

Lead-128k: Demon Copperhead is a complex and resilient character shaped by a life of hardship, neglect, and survival. Born to a drug-addicted mother, he was abandoned in a trailer park in Lee County, Virginia, and nearly died at birth due to his mother's drug use, which led to his being born in the caul—a rare occurrence that his mother claimed was a sign of her own superstitions. He was taken in by the Peggots, who were not his real family but became his de facto parents, giving him a sense of belonging and a nickname, "Demon," that stuck with him as he grew. His early life was marked by instability, including being sent to live with the brutal Crickson family on a farm, where he endured harsh treatment and emotional abuse. He also experienced foster care with Miss Barks, who became a maternal figure but eventually left for a better-paying job as a teacher. Demon's life took a tragic turn when his mother died in an overdose, leaving him in foster care again and further alienated. He later ran away to Nashville, where he encountered a series of misfortunes, including theft, abuse, and being caught in a pawnshop with a hooker. Eventually, he found refuge with his grandmother Betsy Woodall, who had long been estranged from him but now took him in after learning of his mother's death. Demon is marked by his red hair and blue eyes, traits he inherited from his father—a Melungeon man he never knew but who left an indelible mark on his identity. He is also defined by his survival instincts, having learned to navigate the harsh realities of poverty, neglect, and abuse.

Lead-128k + guided-QA (GRPO): Demon Copperhead, the protagonist of the novel, is a resilient and determined young man who navigates a life marked by hardship, neglect, and a complex web of relationships. Born to an addicted mother, he faces a difficult start in life, including a traumatic birth and early experiences with foster care. His early years are shaped by the Peggots, who become a significant part of his life, though their relationship is strained and complicated. He develops a close friendship with Maggot, another foster child, and later forms a bond with Fast Forward, who becomes a mentor and friend. Demon's nickname "Demon" is given to him by others, reflecting his troubled past and the stigma he carries. His relationship with Stoner, his stepfather, is fraught with tension and abuse, leading to a confrontation that results in Demon's injury and eventual placement in foster care. He is placed with Mr. Crickson, a harsh farmer who runs a slave-boy farm, but eventually finds a more supportive home with Coach Winfield and his family in Jonesville. Demon's personality is marked by his determination to prove his worth despite his circumstances, as well as his artistic talent for drawing superheroes and comic strips. He has a complex relationship with his biological mother, who dies from an overdose, and his biological father, whose death he never fully understands. His journey includes running away from home, stealing a tree for Christmas, and eventually finding love and support with Dori, who becomes his girlfriend and later his fiancée. Demon's story is one of resilience, as he struggles with addiction but ultimately seeks redemption through his art and relationships. His character is defined by his search for belonging and self-worth in a world that often overlooks him.

Goldstandard Description: Demon Copperhead is the eponymous, first-person narrator and protagonist of the novel. Because the novel's narrative voice deals with both the events of Demon's life and his reflection on those events, the novel is structured like a memoir. It is also a bildungsroman in which Demon endures several life-altering conflicts and eventually finds peace with himself. The world has been cruel to Demon, and his early life demonstrates The Failure of Society to Protect Its Children. He was born without a father, and his mother struggled with addiction throughout her short life. He navigates several different forms of false families, such as life with his abusive stepfather, placements in two abusive foster homes, and rejection from his father's mother. The lack of love in Demon's childhood and early adolescence makes him radically self-reliant, but it also makes him distrustful of the people who do provide him with security and care. Demon is independent because he must be to survive, but he looks for people to believe in—often misguidedly, as with Fast Forward and Dori. Similarly, he has an attachment to Lee County because it is a space he at least understands, even if it keeps bringing him down. Demon eventually learns that he must focus on himself to receive real love from other people. [truncated]

Fact	Lead-128k	Lead-128k guided-QA
Demon Copperhead is the eponymous narrator of the novel.	False	True
Demon Copperhead is the first-person narrator of the novel.	False	False
Demon Copperhead is the protagonist of the novel.	False	True
The narrative voice of the novel "Demon Copperhead" deals with the events of Demon's life.	False	True
The narrative voice of the novel "Demon Copperhead" includes Demon's reflection on those events.	False	False
The novel "Demon Copperhead" is structured like a memoir.	False	False
"Demon Copperhead" is a bildungsroman.	False	True
In "Demon Copperhead," Demon endures several life-altering conflicts.	True	True
In "Demon Copperhead," Demon eventually finds peace with himself.	False	False
The world has been cruel to Demon Copperhead.	True	True
Demon's early life demonstrates The Failure of Society to Protect Its Children.	True	True
Demon was born without a father.	True	False
Demon's mother struggled with addiction throughout her short life.	False	True
Demon navigates several different forms of false families.	True	True
Demon lived with an abusive stepfather.	False	True
Demon was placed in two abusive foster homes.	True	True
Demon experienced rejection from his father's mother.	False	False
The lack of love in Demon's childhood made him radically self-reliant.	True	True
[Facts truncated]		

Table 15: Example of generated descriptions in CroSS dataset with QA-guided Lead-128k and the corresponding baseline. We demonstrate facts evaluated based on the PRISMA-recall metric.

Character: Demon Copperhead **Book:** Demon Copperhead

Reasoning for 1st Chunk:

Q1: What is Demon Copperhead's role in the story?

E1: Demon Copperhead is the protagonist and narrator of the story, experiencing various challenges and relationships throughout his life.

A1: Protagonist T1: Role

Q2: Who is Demon Copperhead's mother?

E2: Demon Copperhead's mother is a central character in the story, struggling with addiction and neglect.

A2: Addicted mother T2: Relationship

Q3: What is Demon Copperhead's personality like?

E3: Demon Copperhead is portrayed as a resilient and determined character, despite facing numerous hardships.

A3: Resilient and determined T3: Personality

Q4: What event leads to Demon Copperhead's birth?

E4: Demon Copperhead is born to a mother who is struggling with addiction, leading to a difficult start in life.

A4: Difficult birth T4: Event

Q5: Who is Demon Copperhead's foster family?

E5: Demon Copperhead is raised by the Peggots, who become a significant part of his life.

A5: The Peggots T5: Relationship

Q6: What is Demon Copperhead's relationship with Maggot?

E6: Demon Copperhead and Maggot are close friends who grow up together in the same community.

A6: Close friend T6: Relationship

Q7: What is Demon Copperhead's nickname?

E7: Demon Copperhead is given the nickname "Demon" by others, which becomes a part of his identity.

A7: Demon T7: Personality

Q8: What event leads to Demon Copperhead's time with the Peggots?

E8: Demon Copperhead spends time with the Peggots after his mother's struggles with addiction and his own difficult circumstances.

A8: Time with Peggots T8: Event

Q9: What is Demon Copperhead's relationship with Stoner?

E9: Demon Copperhead has a complex relationship with Stoner, who becomes his stepfather and has a significant impact on his life.

A9: Stepfather T9: Relationship

Q10: What event leads to Demon Copperhead's conflict with Stoner?

E10: Demon Copperhead conflicts with Stoner due to Stoner's strict parenting style and his own struggles with identity and self-worth.

A10: Conflict with Stoner T10: Event

Reasoning for 2st Chunk:

Q11: What event leads to Demon Copperhead being placed in foster care?

E11: Demon Copperhead's mother becomes addicted to drugs and is involved in a violent relationship, leading to his placement in foster care.

A11: Mother's addiction and abuse. T11: Event

Q12: Who becomes Demon Copperhead's foster father?

E12: Demon Copperhead's foster father is Mr. Crickson, a farmer who takes in foster children.

A12: Mr. Crickson. T12: Relationship

Q13: What event causes Demon Copperhead to run away from home?

E13: Demon Copperhead runs away from home after a violent confrontation with his stepfather, Stoner.

A13: Confrontation with Stoner. T13: Event

Q14: What is Demon Copperhead's attitude towards his foster family?

E14: Demon Copperhead has a complex attitude towards his foster family, initially feeling out of place but gradually forming bonds with some members.

A14: Complex attitude. T14: Personality

Q15: What event leads to Demon Copperhead being placed in a temporary foster home?

E15: Demon Copperhead is placed in a temporary foster home after his mother is hospitalized and unable to care for him.

A15: Mother's hospitalization. T15: Event

[Trace truncated]

Table 16: Example of generated reasoning trace in CroSS dataset. The trace is from the GRPO-trained Qwen-3-8b reasoning model using Lead-128k.

Character: John Montraville **Book:** Charlotte Temple

Lead-128k: John Montraville is a young, impetuous soldier who becomes infatuated with Charlotte Temple, a pure and innocent girl, and secretly courts her despite his duty and honor. He is torn between his love for Charlotte and his sense of responsibility, but his passion leads him to elope with her, only to discover she was deceived by Mademoiselle La Rue. Montraville’s heart breaks upon learning of Charlotte’s betrayal and her subsequent suffering, ultimately leading to his tragic downfall and death, which he brings upon himself in a fit of vengeance against Belcour.

Fact

John Montraville is a young soldier.	True
John Montraville is an impetuous soldier.	False
John Montraville becomes infatuated with Charlotte Temple.	True
Charlotte Temple is a pure and innocent girl.	False
John Montraville secretly courts Charlotte Temple despite his duty.	False
John Montraville secretly courts Charlotte Temple despite his honor.	False
John Montraville is torn between his love for Charlotte Temple and his sense of responsibility.	False
John Montraville’s passion leads him to elope with Charlotte Temple.	False
John Montraville discovers that Charlotte Temple was deceived by Mademoiselle La Rue.	False
Montraville’s heart broke upon learning of Charlotte’s betrayal.	False
Montraville’s heart broke upon learning of Charlotte’s subsequent suffering.	False
Montraville’s heart breaking led to his tragic downfall.	False
Montraville’s tragic downfall led to his death.	False
Montraville brought his death upon himself in a fit of vengeance against Belcour.	False

Lead-128k + guided-QA: John Montraville is a Lieutenant in the army, an ardent and impetuous soldier who becomes infatuated with Charlotte Temple, a young, innocent girl he meets at a boarding school. He secretly courts her, leading her to elope with him, but later discovers her relationship with Belcour, his friend, which causes his heartbreak. Despite his initial honor and compassion, he succumbs to passion, abandons Charlotte, and marries Julia Franklin. His guilt over betraying Charlotte and the death of her child lead to a tragic end, as he kills Belcour in a fit of rage and dies shortly after, haunted by remorse.

Fact

John Montraville is a Lieutenant in the army.	False
John Montraville is an ardent soldier.	False
John Montraville is an impetuous soldier.	False
John Montraville became infatuated with Charlotte Temple.	True
Charlotte Temple is a young and innocent girl.	False
John Montraville met Charlotte Temple at a boarding school.	False
John Montraville secretly courted Charlotte Temple.	False
John Montraville led Charlotte Temple to elope with him.	False
John Montraville later discovered Charlotte Temple’s relationship with Belcour.	False
Belcour is a friend of John Montraville.	True
John Montraville’s discovery of Charlotte Temple’s relationship with Belcour caused him heartbreak.	False
John Montraville initially displayed honor and compassion.	False
John Montraville succumbed to passion.	True
John Montraville abandoned Charlotte Temple.	True
John Montraville married Julia Franklin.	True
John Montraville felt guilt over betraying Charlotte Temple.	True
John Montraville felt guilt over the death of Charlotte Temple’s child.	False
John Montraville killed Belcour in a fit of rage.	False
John Montraville died shortly after killing Belcour.	False
John Montraville was haunted by remorse after his actions.	True

Goldstandard Description: A charming, dashing young officer in the British Army, John Montraville falls in love with Charlotte Temple in Chichester and hopes to marry her. However, his father refuses to endorse a marriage to a poor woman, so Montraville makes Charlotte his mistress and brings her to America with him. He tries to treat her honorably but is easily deceived by his friend Belcour, who ruins his relationship with Charlotte so he can have her as his own mistress. Despite his inability to control his womanizing impulses, Montraville feels guilty about the misfortune he causes Charlotte and others.

Table 17: Examples of generated descriptions in BookWorm dataset with QA-guided Lead-128k and the corresponding baseline. We demonstrate the facts evaluated based on PRISMA-precision metric.

Character: John Montraville **Book:** Charlotte Temple

Reasoning for 1st Chunk:

Q1: What role does John Montraville play in the story?

E1: John Montraville is a major character in the story, a young soldier who becomes involved with Charlotte Temple.

A1: Major character T1: Role

Q2: Who is John Montraville related to in the story?

E2: John Montraville is related to Charlotte Temple, as he becomes romantically involved with her.

A2: Charlotte Temple T2: Relationship

Q3: What is John Montraville's personality like?

E3: John Montraville is portrayed as a tender, eloquent, and ardent lover, but also impetuous and driven by his passions.

A3: Ardent and impetuous T3: Personality

Q4: What event leads John Montraville to meet Charlotte Temple?

E4: John Montraville meets Charlotte Temple when he encounters her and her governess at a church in Chichester.

A4: Meeting Charlotte Temple T4: Event

Q5: What decision does John Montraville make regarding his relationship with Charlotte Temple?

E5: John Montraville decides to elope with Charlotte Temple, despite knowing their families would not approve.

A5: Decide to elope T5: Event

Q6: What is John Montraville's profession?

E6: John Montraville is a soldier, as he is mentioned as a Lieutenant in the army.

A6: Soldier T6: Role

Q7: What is John Montraville's relationship with Belcour?

E7: John Montraville is friends with Belcour, who accompanies him on his visits to Charlotte Temple.

A7: Friend T7: Relationship

Q8: What is John Montraville's attitude towards marriage?

E8: John Montraville is initially cautious about marriage, as his father warns him against rushing into it.

A8: Cautious T8: Personality

Q9: What is one of the events involving John Montraville and Charlotte Temple?

E9: One event involving John Montraville and Charlotte Temple is their clandestine meetings at a summer-house.

A9: Clandestine meetings T9: Event

Q10: What is John Montraville's relationship with his father?

E10: John Montraville has a respectful relationship with his father, who gives him advice about marriage and duty.

A10: Respectful T10: Relationship

Reasoning for 2st Chunk:

Q11: Who is John Montraville romantically involved with?

E11: John Montraville is romantically involved with Charlotte Temple and Julia Franklin.

A11: Charlotte Temple T11: Relationship

Q12: What event leads to John Montraville's conflict with Charlotte Temple?

E12: John Montraville's conflict with Charlotte Temple arises from her elopement and his subsequent discovery of her relationship with Belcour.

A12: Elopement T12: Event

Q13: What event leads to John Montraville's conflict with Julia Franklin?

E13: John Montraville's conflict with Julia Franklin arises from his indecision about marrying her while still being involved with Charlotte Temple.

A13: Indecision T13: Event

Q14: What event leads to John Montraville's discovery of Charlotte's infidelity?

E14: John Montraville discovers Charlotte's infidelity when he finds her in bed with Belcour.

A14: Discovery T14: Event

Q15: What is John Montraville's role in the story's resolution?

E15: John Montraville plays a significant role in the story's resolution by ultimately leaving Charlotte and pursuing a relationship with Julia Franklin.

A15: Pursuing relationship T15: Event

Q16: What is John Montraville's personality trait regarding honor?

E16: John Montraville is portrayed as having a strong sense of honor, which conflicts with his romantic entanglements.

A16: Strong sense of honor T16: Personality

[Trace truncated]

Table 18: Example of generated reasoning trace in BookWorm dataset. The trace is from the GRPO-trained Qwen-3-8b reasoning model with Lead-128k as input context.