IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 13 December 2025, accepted 26 January 2026, date of publication 28 January 2026, date of current version 3 February 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3659054

RESEARCH ARTICLE

LLM QLoRA Fine-Tuning of Llama, DeepSeek, and Qwen: A Skyrim Case Study

MARCOS EDUARDO PIVARO MONTEIRO, MARCOS TALAU, AND HEITOR SILVÉRIO LOPES
Graduate Program in Electrical and Computer Engineering (CPGEI-CT), Federal University of Technology—Paraná (UTFPR), Curitiba, State of Paraná 80230-901, Brazil

Corresponding author: Marcos Eduardo Pivaro Monteiro (marcose@utfpr.edu.br)

This work was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-CAPES (ROR identifier: 00x0ma614) for the Article Processing Fee.

ABSTRACT Fine-tuning Large Language Models (LLMs) for domains that demand extensive background knowledge, such as video game lore, involves navigating trade-offs between model architecture, scale, and the organization of training data. In this study, we examine these factors through the use of Quantized Low-Rank Adaptation (QLoRA) applied to the lore of Skyrim®. Our analysis considers nine models from the DeepSeek, Llama, and Qwen families at three parameter scales (∼8B, ∼13B, and ∼33B). Each model was fine-tuned on unstructured, structured, and summarized datasets using LoRA ranks of 16, 32, and 64. Performance was evaluated with standard metrics (Perplexity, ROUGE, BLEU), a robust ensemble qualitative LLM-as-a-Judge framework, and a dedicated benchmark for catastrophic forgetting. The results show a consistent trade-off: structured datasets produce the most fluent outputs, while summarized datasets tend to improve factual accuracy, typically at the cost of accelerated degradation in general knowledge. Among the model families, Llama performs best at the ∼8B and ∼33B scales, whereas the code-specialized DeepSeek models have an edge at the ∼13B size. Furthermore, our analysis of training dynamics reveals that higher LoRA ranks significantly improve convergence speed and stability. Overall, the optimal trade-off between performance, efficiency, and knowledge retention was achieved with Llama-3.1-8B fine-tuned on a summarized dataset with a LoRA rank of 64.

INDEX TERMS Large language models, LLM fine-tuning, LoRA, DeepSeek, Llama, Qwen, game lore.

I. INTRODUCTION

The rapid growth of Large Language Models (LLMs) marks a major shift in artificial intelligence, with these models demonstrating impressive skill across many general-purpose tasks. As more organizations try to adapt these models for specific purposes [1], a new challenge has become a key research focus: making them work effectively in niche areas that require deep, specialized knowledge. Video game lore, such as that from the expansive universe of *The Elder Scrolls V: Skyrim*®,[1] represents a quintessential example of such a domain. Such a world is defined by an intricate body of text, complete with its own terminology, deep character histories, and complex rule systems, elements that are challenging for general-purpose models to learn with factual consistency.

The conventional method of full fine-tuning, while effective for domain adaptation, is computationally prohibitive for models with billions of parameters, especially on consumer-grade hardware. Full-model fine-tuning requires enormous memory resources for model weights, optimizer states, and gradients, making it inaccessible for many researchers and developers [2]. Such a fundamental limitation has stimulated the development of Parameter-Efficient Fine-Tuning (PEFT) techniques, which aim to achieve comparable performance by updating only a small fraction of the model's parameters [3].

Among the various PEFT methodologies, Low-Rank Adaptation (LoRA) has appeared as a particularly effective

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna D'Ulizia.

[1] Skyrim® is a registered trademark of ZeniMax Media Inc., developed and published by Bethesda Softworks, and is used here under nominative fair use for educational and research purposes. ZeniMax Media Inc. is a subsidiary of Microsoft.

and widely adopted approach for creating specialized models without having the costs of full fine-tuning [4]. LoRA and its variants, such as QLORA for quantized models, have been successfully applied in diverse domains, from enhancing named entity recognition [5] and correcting Text-to-SQL errors [6], and advancing sentiment analysis in low-resource languages [7], [8]. These techniques offer a practical pathway to customizing LLMs for specialized tasks, as demonstrated in various comparative analyses [9], [10].

Despite the widespread adoption of LoRA, several key questions remain regarding its optimal application, particularly concerning the interplay between model architecture, scale, and the nature of the fine-tuning data. This study presents a comprehensive comparative analysis of LoRA fine-tuning applied specifically to the domain of Skyrim lore. We investigate the interplay of several key factors to provide a better understanding of their impact on model performance in a creative, knowledge-intensive context. Our research is guided by the following core questions:

1) How does the performance of fine-tuned models change with increasing parameter counts across different size tiers (∼8B, ∼13B, and ∼32B)?
2) What is the effect of the fine-tuning dataset's structure, comparing models trained on unstructured narrative text versus those trained on structured, attribute-rich data?
3) How do the foundational architectures of leading open-source model families, DeepSeek [11], Llama [12], and Qwen [13], compare when fine-tuned on this specialized task?
4) What is the impact of the LoRA rank, a key hyperparameter, on the model's ability to learn and reproduce domain-specific knowledge?
5) To what extent does fine-tuning on this specialized domain induce catastrophic forgetting of general knowledge, and how does this vary by model architecture and dataset type?

The primary contribution of this work is a systematic, multi-faceted empirical study that provides insights for those seeking to adapt LLMs for specialized creative and knowledge-intensive domains. By isolating and analyzing the effects of model scale, architecture, data format, and fine-tuning configuration, we offer a detailed roadmap for achieving effective domain adaptation within the constraints of accessible hardware. Our findings are relevant to a broad range of applications where domain-specific fine-tuning is critical, including but not limited to game development [14], interactive fiction, and specialized chatbots [15].

This paper is structured as follows: Section II reviews the principles of PEFT, LoRA, and the foundational model architectures under investigation. Section III details our dataset preparation, experimental design, and evaluation framework. Section IV presents the quantitative and qualitative results of our comparative analysis, which are then interpreted in Section V. Finally, Section VI summarizes our findings and suggests directions for future work.

II. BACKGROUND AND RELATED WORK

A. PARAMETER-EFFICIENT FINE-TUNING (PEFT)

As LLMs have grown in size, the computational resources required for full fine-tuning have become a major bottleneck. The process of updating all of a model's weights is not only slow but also memory-intensive, often requiring multiple high-end GPUs [2]. This challenge has motivated the development of a suite of PEFT methods. The core principle of PEFT is to freeze the vast majority of the pre-trained model's parameters and introduce a few new, trainable parameters to adapt the model to a new task [1]. These techniques have proven highly effective, often achieving performance comparable to full fine-tuning while only updating less than 1% of the model's total weights, thereby significantly lowering the barrier to entry for model customization and enabling applications in resource-constrained environments like IoT edge servers [16].

The application of PEFT is broad and varied, with studies demonstrating its effectiveness in fields ranging from legal and insurance domains [1] to smart grids [17] and automated program repair [18]. The efficiency gains offered by PEFT have also made it a central aspect of research into federated learning with LLMs, where minimizing communication overhead is critical [19], [20]. Furthermore, the security of distributed training frameworks for LLMs often relies on isolating PEFT modules within trusted execution environments, highlighting the modularity and importance of these techniques [21].

B. PRINCIPLES OF LOW-RANK ADAPTATION (LoRA)

LoRA is a prominent PEFT technique based on the hypothesis that the change in weights during model adaptation has a low *intrinsic rank* [4]. Instead of directly updating a large, pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, LoRA models the update with two smaller, low-rank matrices, $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$, where the rank $r \ll \min(d, k)$. During training, W_0 remains frozen, and only the parameters of A and B are updated. Using $h = W_0 x$, the modified forward pass is then expressed as [4]

$$h = W_0 x + BAx \tag{1}$$

where h is the output vector and x is the input. This decomposition significantly reduces the number of trainable parameters compared to updating the entire W_0 matrix.

The modularity of LoRA has led to its integration in advanced architectures, such as Mixture-of-Experts (MoE) frameworks, where different LoRA modules can be specialized for different tasks or data distributions [22], [23]. A key hyperparameter in LoRA is the rank (r), which determines the dimensionality of the trainable matrices and thus the expressive capacity of the adaptation.

To make this technique even more accessible, Quantized Low-Rank Adaptation (QLoRA) was introduced [2]. QLoRA further reduces memory usage by quantizing the pre-trained model to 4-bit precision and then fine-tuning the small LoRA

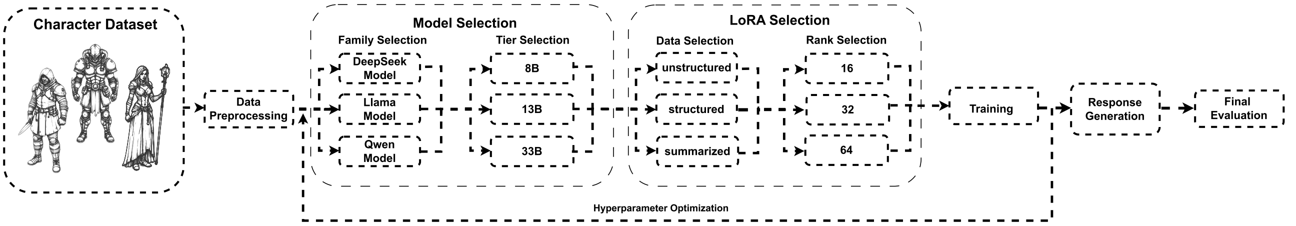


FIGURE 1. **Overview of the proposed workflow, illustrating the complete pipeline from initial character dataset,[2] data preprocessing, model and LoRA selection, training with hyperparameter optimization, response generation, and final evaluation.**

adapters on top of these quantized weights. This approach relies on the 4-bit NormalFloat (NF4) data type, which is information-theoretically optimal for the zero-centered normal distribution of weights typically found in Transformer models. It has been demonstrated [2] that this pipeline maintains performance parity with 16-bit full fine-tuning on complex benchmarks. Furthermore, recent empirical studies have specifically validated the robustness of 4-bit quantization for the architectures employed in this study. For instance, Llama 3 models have been found to maintain competitive performance at 4-bit precision, significantly outperforming smaller full-precision models [24]. Similarly, investigations into the DeepSeek [25] and Qwen [26] families confirm that 4-bit quantization represents a stable operating point, avoiding the sharp performance degradation observed at 3-bit or lower precisions while enabling the efficient deployment of larger, more capable models. This innovation has been instrumental in enabling the fine-tuning of very large models on single, consumer-grade GPUs, democratizing the ability to create specialized LLMs [27].

Despite its widespread success, LoRA is not without limitations. A key challenge identified in recent literature is the *accuracy plateau*, a phenomenon where simply increasing the LoRA rank (r), and thus the number of trainable parameters, fails to bridge the performance gap between LoRA and standard full fine-tuning [28]. This suggests that the low-rank nature of the update matrix, rather than just the number of parameters, can become an inherent bottleneck, limiting the expressiveness required for complex adaptation tasks. This raises critical questions about the trade-offs between parameter efficiency and the absolute performance ceiling of LoRA, motivating our empirical investigation into the effects of varying the LoRA rank in Section IV.

C. FOUNDATIONAL MODEL ARCHITECTURES

The models investigated in this study, Llama [12], Qwen [13], and DeepSeek [11], are all based on the decoder-only Transformer architecture, which has become the de facto standard for modern LLMs. This architecture is a significant evolution from the original encoder-decoder Transformer [33] due to its efficiency in autoregressive text generation. Although they share this core design, each model family incorporates unique modifications and design philosophies that differentiate its performance and capabilities.

1) LLAMA

The Llama family of models, developed by Meta, has been highly influential in the open-source community. Models like Llama 2 [12] and its successors are powerful dense architectures that have consistently pushed the boundaries of performance for publicly available models. They incorporate architectural optimizations such as Grouped-Query Attention (GQA) to improve inference efficiency without significant performance degradation. The widespread availability and strong performance of the Llama architecture have made it a frequent subject of LoRA fine-tuning studies [3].

2) QWEN

The Qwen series of models from Alibaba Cloud is another family of high-performance, dense transformer architectures. Recent iterations like Qwen2 have demonstrated state-of-the-art performance on a wide range of benchmarks and also utilize optimizations like GQA, making them highly competitive with other leading open-source models [13].

3) DeepSeek

The DeepSeek family of models, including its Coder variants, is distinguished by its pre-training on a massive corpus of both natural language and source code [11], [31]. These models are specifically designed to excel at reasoning and logical tasks, which makes their adaptation to a creative, narrative-driven domain like Skyrim lore a particularly interesting case study in transfer learning from a code-centric foundation to a purely linguistic one. The fine-tuning of DeepSeek models with LoRA has been a subject of recent interest, highlighting the community's efforts to adapt these powerful models for a variety of tasks [10].

III. METHODOLOGY

A. SYSTEM MODEL

The overall workflow of our fine-tuning and evaluation solution is presented in Fig. 1. The pipeline begins with an initial character dataset, which undergoes preprocessing to create three distinct versions for fine-tuning: an unstructured dataset for general style adaptation, a structured dataset for

[2] The character illustrations used in this diagram are for artistic and illustrative purposes only and are not official assets or characters from *The Elder Scrolls V: Skyrim.*

TABLE 1. Selected instruction-tuned models for comparative analysis via 4-bit QLoRA.

Size Tier	DeepSeek	Llama	Qwen
~8B	`deepseek-llm-7b-chat` [11]	`Llama-3.1-8B-Instruct` [29]	`Qwen3-8B` [13]
~13B	`d-coder-v2-lite-instruct` (16B) [30]	`Llama-2-13B-chat-hf` [12]	`Qwen3-14B` [13]
~33B	`deepseek-coder-33B-instruct` [31]	`CodeLlama-34b-Instruct-hf` [32]	`Qwen3-32B` [13]

attribute extraction, and a summarized dataset for abstractive summarization.

Following data preparation, the process moves to the Model and LoRA selection stages. This involves a systematic selection of foundational models from three prominent architectural families, DeepSeek, Llama, and Qwen, across three parameter tiers (~8B, ~13B, and ~33B). Concurrently, the specific fine-tuning configuration is chosen, which includes selecting one of the three preprocessed datasets and a LoRA rank of 16, 32, or 64.

The selected models and LoRA configurations then enter the Training stage, which runs the hyperparameter optimization loop. After training, the models move to the Response Generation stage, where we prompt each unique configuration to generate responses for a test set of characters. These generated responses are then used in the Final Evaluation to calculate both the quantitative metrics (e.g., perplexity, ROUGE) and to serve as input for the qualitative LLM-as-a-Judge framework. The results from this comprehensive analysis are used to determine the optimal configuration.

B. DATASET PREPARATION

The foundation of this study is a dataset of characters from the video game *The Elder Scrolls V: Skyrim*. The initial data was sourced from the Mantella project,[3] which provides a collection of character biographies. This raw data, with a total of 2,746 characters, forms the basis for our unstructured dataset. Unlike traditional generalization tasks, our objective was to inject specific domain knowledge; therefore, the training process was designed to expose the models to the entire corpus via the data cycling strategy detailed in Section III-E. For the response generation and final evaluation phase, we utilized a representative subset of 269 characters (≈10% of the total) to assess the model's ability to recall and reproduce the learned attributes and narratives.

To facilitate a more rigorous analysis of the models' reasoning and data extraction capabilities, a structured dataset was created. This was achieved by augmenting the original unstructured data with several new fields, including the character's home city, race, gender, and current location. These additional fields were populated by integrating ground-truth metadata from the "People of Skyrim" database [34]. We injected a standardized key-value header before each biography, creating a format designed to test the model's ability to learn and reproduce rigid schema constraints.

A third dataset, the summarized dataset, was created to investigate the impact of data density on fine-tuning effectiveness. This version was synthetically generated using Google's Gemini to condense the original biographies into high-entropy paragraphs (maximum 3 sentences). The system prompt explicitly instructed the model to strip away stylistic filler and focus strictly on core attributes like role, location, and relationships. The generation scripts for this process are available in the project repository [35].

C. HARDWARE AND SOFTWARE ENVIRONMENT

The experiments were executed on two distinct servers to accommodate the different model scales. The models were trained on a high-performance computing server equipped with dual Intel® Xeon® Gold 6338 CPUs, providing a total of 128 logical processors. This system is configured with 512 GB of DDR4 RAM in a Non-Uniform Memory Access (NUMA) architecture. For GPU acceleration, it is fitted with four NVIDIA L40S graphics cards, each providing 48 GB of GDDR6 memory and support for CUDA 12.

The second server, used for debugging, is equipped with dual Intel® Xeon® E5-2697 v3 CPUs, totaling 56 logical processors. It is configured with 110 GB of RAM, also in a NUMA architecture. GPU acceleration on this machine is provided by a heterogeneous setup consisting of one NVIDIA GeForce RTX 4060 Ti (16 GB) and two NVIDIA GeForce RTX 3060 (12 GB) graphics cards, also supporting CUDA 12.

The software stack was built using Python 3.11 and the PyTorch 2.8.0 deep learning framework. Key libraries from the Hugging Face ecosystem were utilized, including `transformers` (v4.53.0) for model and tokenizer access, `peft` (v0.15.2) for implementing LoRA, `trl` (v0.19.0) for the supervised fine-tuning trainer, and `bitsandbytes` (v0.47.0) for 4-bit quantization. A complete list of dependencies, configuration files, and reproduction scripts is available in the project repository [35].

D. MODELS UNDER TEST

To conduct a robust cross-architectural analysis, models were selected from three prominent families: DeepSeek, Llama, and Qwen. The selection spans three approximate parameter size tiers, ~8B, ~13B, and ~33B, to evaluate performance scaling on consumer-grade hardware. This study employs a consistent 4-bit QLoRA fine-tuning process for all experiments. Instead of using pre-quantized model variants, we load the official full-precision instruction-tuned

[3] https://github.com/art-from-the-machine/Mantella

models and apply quantization on-the-fly via the ‘bitsandbytes‘ library, as managed by the Unsloth framework. This ensures a standardized and scientifically rigorous comparison of the models’ architectural performance under identical quantization conditions.

A key methodological decision was to exclusively use models that have undergone instruction or chat fine-tuning. This ensures a fair comparison focused on architectural differences rather than discrepancies between base and tuned model capabilities. Furthermore, *distilled* models were deliberately excluded to compare foundational architectures directly. The selected models are detailed in Table 1.

The model selection in each tier acknowledges the realities of the current open-source landscape:

1) ∼8B TIER

This tier represents the most scientifically sound basis for architectural comparison in this study. All three models, `deepseek-llm-7b-chat` (DeepSeek-7B) [11], `Llama-3.1-8B-Instruct` (Llama-3.1-8B) [29], and `Qwen3-8B` (Qwen3-8B) [13], are from their respective latest generations, are instruction-tuned, and possess comparable parameter counts. This allows for a controlled analysis of foundational architectural differences.

2) ∼13B TIER

This tier requires acknowledging a significant confounding variable, as the Llama 3 family lacks a model in this parameter range. This necessitates using the previous generation’s `Llama-2-13B-chat-hf` model (Llama-2-13B) [12]. Furthermore, DeepSeek’s closest available model is the `deepseek-coder-v2-lite-instruct` (D-Coder-V2-16B), which is heavily specialized for code generation [30]. The Qwen model, `Qwen3-14B` (Qwen3-14B), is from the third generation [13]. Consequently, this experiment is framed as a generational and domain-specialization comparison.

3) ∼33B TIER

To ensure representation from the Llama family and due to the lack of alternatives at this tier, we include `CodeLlama-34b-Instruct-hf` (CodeLlama-34B) [32]. This model is architecturally based on Llama 2 but has undergone extensive pre-training on source code. Its inclusion allows for a three-way comparison against `deepseek-coder-33B-instruct` (D-Coder-33B) [31] and `Qwen3-32B` (Qwen3-32B) [13]. This setup provides a unique opportunity to analyze not only architectural differences at scale but also the transfer-learning capabilities of domain-specialized models on a narrative task.

It is important to acknowledge that this selection, driven by the availability of models in the open-source landscape, introduces unavoidable confounding variables in the ∼13B and ∼33B tiers. Specifically, the inclusion of a previous-generation Llama 2 model and several code-specialized models means that comparisons in these tiers assess performance across different generational baselines and pre-training objectives, rather than solely isolating architectural features. This context is crucial for interpreting the results, which are further discussed in Section V.

TABLE 2. Comprehensive experimental configuration and hyperparameters. This table summarizes the fixed settings applied across all experiments and defines the variable parameters for specific runs.

Category	Configuration / Value
Models (Size)	DeepSeek-7B (7B), Llama-3.1-8B (8B), Qwen3-8B (8B), D-Coder-V2 (16B), Llama-2-13B (13B), Qwen3-14B (14B), D-Coder-33B (33B), CodeLlama-34B (34B), Qwen3-32B (32B)
Quantization	4-bit NormalFloat (NF4) via `bitsandbytes`
LoRA Rank (r)	Variable: $\{16, 32, 64\}$
LoRA Alpha (α)	$2 \times r$ (i.e., $\{32, 64, 128\}$)
Target Modules	`q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj`
Optimizer	`paged_adamw_8bit`
Learning Rate	1×10^{-4} (Peak)
LR Scheduler	Cosine Decay
Batch Size	Effective: 128 (Gradient Accumulation: Auto)
Max Seq. Length	2048 tokens
Epochs	Early Stopping (Patience: 4 evals $\approx$ 4.4 epochs)
Random Seeds	1003722, 9224736, 4315468
Hardware	NVIDIA L40S (48GB VRAM) × 1
Frameworks	PyTorch 2.8.0, Transformers 4.53.0, PEFT 0.15.2, TRL 0.19.0

E. TRAINING HYPERPARAMETER SELECTION

To ensure a controlled and comparable experimental environment, key training and fine-tuning hyperparameters were established. A summary of these common parameters is provided in Table 2. The computational cost varied significantly by model size: on the specified hardware, training a single stage took approximately 6 hours for the ∼8B models, 12 hours for the ∼13B models, and 24 hours for the ∼33B models. Notably, the inference phase for response generation required a duration comparable to the training phase. Consequently, when accounting for the full experimental matrix (encompassing all architectures, dataset formats, and LoRA ranks) across three independent random seeds, the total computational expenditure for training and generation exceeded 6,800 GPU-hours. In contrast, the subsequent evaluation phase was computationally negligible due to the efficient batching of the judge models.

1) LoRA CONFIGURATION

For all experiments, we employed the QLoRA methodology, fine-tuning the models in 4-bit precision. The selection of LoRA-specific hyperparameters, alpha (α) and target modules, was guided by established best practices.

A common and effective heuristic is to set the LoRA alpha (α) as double the rank (r). This configuration gives significant weight to the LoRA updates without requiring an excessively high rank. Thus, we adopted the setting $\alpha = 2 \times r$ for all fine-tuning runs. For all models, the target modules were set to adapt all major linear layers within the Transformer architecture: the query (‘q_proj‘), key (‘k_proj‘), value (‘v_proj‘), and output (‘o_proj‘) projections of the self-attention mechanism, as well as the linear layers of the feed-forward networks (‘gate_proj‘, ‘up_proj‘, ‘down_proj‘). Targeting this comprehensive set of layers ensures that the adaptation is not limited to the attention mechanism but also influences the model’s feed-forward networks, providing a more holistic fine-tuning process.

The LoRA rank (r) itself is treated in this work as a primary experimental variable, with values of 16, 32, and 64 being tested to investigate its impact on performance, as detailed in Section IV.

2) MULTI-STAGE DATA CYCLING STRATEGY

A standard 80/20 train/validation split, while effective for preventing overfitting, presents a unique challenge for knowledge-intensive fine-tuning. Our objective is the factual recall of the entire character corpus; hiding 20% of the data in a validation set permanently prevents the model from learning those specific facts. To address this, we employed a multi-stage, data-cycling ensemble approach [35] implemented in two phases:

1) *Initial Adaptation:* Three distinct models were initialized with different random seeds and trained on three unique 80/20 data splits. We utilized early stopping with a patience of 4 evaluation intervals (approximately 4.4 epochs) based on validation metrics calculated on the held-out 20% partition, retaining the model checkpoint with the lowest validation metric to ensure a robust, non-overfit baseline.
2) *Continued Training (Cycling):* We performed a second training phase where each model from the first stage was exposed to a data partition from a different split configuration following a strict cyclic permutation (i.e., Split 1 → Split 2, Split 2 → Split 3, and Split 3 → Split 1).

This approach ensures that every model is eventually exposed to 100% of the available character data in a controlled manner, solving the ‘‘unseen fact’’ problem while maintaining the regularization benefits of validation-based early stopping. Preliminary experiments indicated that this two-stage workflow offered greater stability compared to single-stage training, preventing premature convergence. Finally, performance metrics were averaged across these three runs to smooth out variance. Further reproduction details are available at [35].

F. MODEL-SPECIFIC PROMPT FORMATTING

A critical aspect of instruction fine-tuning is adhering to the specific chat or instruction template for which each model was optimized. Using a model’s native format is essential for achieving optimal performance. Therefore, our methodology incorporates model-aware prompt formatting, which is applied based on the selected model family. All data samples were converted into the appropriate format using the chat template function provided by the Hugging Face Transformers library.

For the unstructured dataset, which consists of simple character biographies, the format is a straightforward instruction-response pair. The character’s name is used to form a user instruction (e.g., ‘‘Provide a detailed description of…’’), and the biography serves as the assistant’s response.

For the structured dataset, the formatting is adapted to train the models on attribute extraction and structured data generation. The instruction prompts the model to identify a character’s key attributes (e.g., home city, race, gender, location) from their lore. The target response is a structured key-value list of these attributes, followed by the full biography. This approach is designed to fine-tune the model’s ability to parse text and reorganize key information into a specified, structured format.

Finally, for the summarized dataset, the prompt format mirrors that of the structured version to directly test the model’s ability to extract information from a more condensed source. The instruction again prompts the model to identify the character’s key attributes. The target response is a structured key-value list of these attributes, but it is followed by the concise, summarized biography instead of the full-length one. This setup is designed to evaluate how data density impacts the model’s ability to learn and reproduce factual information when the narrative context is less verbose.

G. EXPERIMENTAL SETUP

To evaluate the performance of the fine-tuned models, a comprehensive experimental matrix was designed. The generation of results iterates through each combination of model, dataset type, and LoRA rank. For each fine-tuned model, a standardized prompt was used to query information about a list of characters from the test set. The prompt specifically instructs the model to return a structured, JSON-like output containing the character’s key attributes, thereby testing its ability for structured data extraction post-fine-tuning. The response generation pipeline primarily utilized deterministic decoding (greedy search) to ensure reproducibility. However, to mitigate failures where models initially outputted invalid JSON structures, a fallback mechanism was implemented: if the deterministic pass failed, a second attempt was made using non-deterministic sampling. This fallback was triggered for exactly 10.51% of the test cases. Importantly, this mixed approach does not compromise the comparability of the positive results. Our analysis indicates that responses generated via the non-deterministic fallback consistently yielded lower factual accuracy and quality scores compared to successful deterministic outputs, often introducing hallucinations. Consequently, the inclusion of these non-deterministic samples ensures experiment completeness without artificially inflating the performance metrics.

1) EXPERIMENT 1: IMPACT OF MODEL SCALE

To assess the effect of model size on fine-tuning performance, models were evaluated across three distinct parameter tiers: ∼8B, ∼13B, and ∼33B. By comparing the performance

System Prompt for LLM-as-a-Judge

You are an expert evaluator of Large Language Models, specializing in creative writing and game lore analysis. Your task is to compare the following responses to a specific question about the game Skyrim. Each response is provided in a structured format containing ’character_name’, ’home_city’, ’race’, ’gender’, ’location’, and ’biography’. Evaluate each response based on the following criteria:

1) **Factual Accuracy of ’Home City’, ’Race’, ’Gender’, ’Location’:** Compare these fields to the ground truth provided. Mark ’Correct’ or ’Incorrect’ for each.
2) **Quality of ’Biography’:** How well does the ’biography’ description align with the character’s overall biography and established Skyrim lore, and how well does it address the part of the question related to biography? Assess its detail, accuracy, and coherence.
3) **Overall Response Coherence and Completeness:** Does the entire response make sense? Are all requested fields present and well-formatted? Is the tone and language consistent with Elder Scrolls lore and the ground truth?
4) **Overall Best Response:** Which model provides the most accurate, coherent, and complete response overall for this character?

-- Ground-Truth Character Data --
[Ground Truth Data Here]
-- End Ground-Truth --
The original question asked was: "[Original Question Here]"
Here are the responses from the models: [Model Responses Here]
For each model’s response, indicate ’Correct’ or ’Incorrect’ for ’Home City’, ’Correct’ or ’Incorrect’ for ’Race’, ’Gender’, and ’Location’ based on the Ground-Truth. Then, provide a biography quality score for each model’s response based on the quality, correctness based on ground truth, and coherence of the ’Biography’, and also provide the overall coherence and quality. Finally, provide an overall ranking of the models from best to worst based on all criteria. Your final output MUST be a single JSON object with the following keys: ’model_evaluations’: (list of objects, one for each model, with keys ’model_key’, ’home_city_correct’, ’race_correct’, ’gender_correct’, ’location_correct’, ’biography_quality_score’ (1-5), ’overall_ranking’: (list of model keys, ordered from best to worst).
JSON Output:

FIGURE 2. The exact system prompt used for the LLM-as-a-Judge evaluation, detailing the role, criteria, and required output format.

of models from the same architectural family across these tiers, this experiment aims to quantify the improvements in lore accuracy and structured data generation that result from increased model scale.

2) EXPERIMENT 2: EFFICACY OF DATA FORMATTING

This experiment directly measures the impact of the fine-tuning data’s structure. For every model and LoRA rank configuration, three separate versions were trained: one on the unstructured (biography only) dataset, one on the structured (attribute-rich) dataset, and one on the summarized dataset. By comparing the performance of these three versions on the final evaluation task, we can determine the extent to which providing structured, explicit data during fine-tuning enhances the model’s ability to perform structured data extraction.

3) EXPERIMENT 3: LoRA RANK ANALYSIS

To understand the sensitivity of fine-tuning to the capacity of the adapter, each experiment was conducted using three different LoRA ranks (r): 16, 32, and 64. By comparing the results for each rank, this analysis investigates the trade-off between the number of trainable parameters in the LoRA adapter and the resulting model performance, providing insight into the optimal configuration for this specific domain adaptation task.

4) EXPERIMENT 4: CROSS-ARCHITECTURAL PERFORMANCE ANALYSIS

Within each size tier, a direct comparison of models from the DeepSeek, Llama, and Qwen families was conducted. This analysis is designed to highlight the inherent strengths and weaknesses of each foundational architecture when applied to the specific domain of game lore. By keeping the model size, training data, and fine-tuning parameters constant, this experiment isolates the architectural differences as the primary variable.

5) EXPERIMENT 5: CATASTROPHIC FORGETTING EVALUATION

To quantify the risk of catastrophic forgetting, where the acquisition of new, specialized knowledge degrades the model’s pre-existing capabilities, we conducted a benchmark evaluation using the TriviaQA dataset [36]. To ensure statistical robustness, we constructed three distinct test sets, each containing 100 randomly sampled general knowledge questions (totaling 300 unique queries). All fine-tuned models were evaluated against these sets to measure the retention of general knowledge relative to their respective base models. The generation process was set to deterministic (greedy decoding) to ensure reproducibility, utilizing exact matching and normalized string comparison for verification. This experiment is critical for determining whether high-density training data, such as the summarized dataset, induces greater degradation of general reasoning compared to unstructured or structured formats.

H. EVALUATION FRAMEWORK

To provide a comprehensive assessment of model performance, a multi-faceted evaluation framework was implemented, combining automated quantitative metrics with scalable qualitative analysis via an LLM-as-a-Judge. This approach allows for the measurement of both factual recall and qualitative attributes.

1) QUANTITATIVE METRICS

Quantitative metrics were employed to objectively measure model fluency and content overlap against a ground-truth reference.

a: PERPLEXITY (PPL)

Perplexity was calculated for each fine-tuned model to measure its adaptation to the linguistic style of the Skyrim domain. A held-out test set of character biographies was processed by each model, and the cross-entropy loss was used to compute the perplexity score. A lower PPL indicates that the model is less *surprised* by the domain-specific text, signifying a higher degree of fluency and better internalization of the narrative patterns.

b: ROUGE AND BLEU

To quantify factual recall and content similarity, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and BLEU (Bilingual Evaluation Understudy) scores were computed. The *biography* field generated by each model for a given character was compared against the ground-truth biography from the structured dataset. While originally designed for summarization and translation, these metrics serve as a valuable proxy for how effectively each model can reproduce key factual information and narrative elements from its training.

c: GENERAL KNOWLEDGE ACCURACY

To quantify catastrophic forgetting, we employed an accuracy metric based on the TriviaQA benchmark [36]. Unlike the lore-specific metrics, this measures the model's retention of pre-existing world knowledge. The accuracy is calculated as the percentage of exact matches (normalized for case and punctuation) between the generated response and the ground-truth answer across the general knowledge test sets. A significant decrease in this score relative to the base model serves as a direct indicator of knowledge degradation.

2) LLM-AS-A-JUDGE

To conduct a scalable and robust qualitative assessment, we implemented an ensemble LLM-as-a-Judge methodology. To mitigate the potential bias of a single evaluator, we employed two distinct high-performance models: Google's proprietary Gemini (accessed via API) and the open-weights EVA-Qwen2.5-72B [37], a full-parameter finetune specialized in roleplay and creative writing, running locally at a temperature of 0.1 to allow retries while ensuring near-determinism. The evaluation process was performed for each character in the test set, with generated responses presented to both judges. The final qualitative scores reported in this study represent the average of the evaluations provided by these two distinct models, normalized to account for differences in their internal scoring distributions [35], ensuring a more balanced perspective on factual accuracy and narrative quality.

The judge was provided with a comprehensive system prompt designed to enforce a strict evaluation protocol. The prompt explicitly instructed the model to act as an expert evaluator of creative writing and game lore, provided with the user's original question, the candidate model's response, and the full ground-truth character data. As detailed in Fig. 2, the evaluation rubric required binary verification (Correct/Incorrect) for specific attributes (Home City, Race, Gender, Location) and a qualitative assessment of the biography's alignment with established lore. To ensure automated parsing, the judge was constrained to output its reasoning and final ranking in a strict JSON format.

Furthermore, the judge's reasoning and scoring were manually spot-checked on random subsets of the data to validate alignment with the rubric. Additionally, the factual verification process employed a hybrid approach: a deterministic script initially checked for exact string matches against the ground truth to handle obvious successes and failures, deferring complex or ambiguous variations to the LLM judge [35].

To ensure structured and parsable results, the judge was constrained to return its evaluation in a JSON format, containing its detailed reasoning and an overall ranking of the models from best to worst for that specific query. This process was repeated across the entire test set, and the final rankings were aggregated to calculate win-rates and determine the overall qualitative performance of each experimental configuration.

IV. RESULTS AND ANALYSIS

This section presents the empirical findings of our comprehensive experimental matrix. The performance of each fine-tuned model configuration was evaluated using a combination of quantitative metrics (Perplexity, ROUGE, BLEU) and qualitative scores derived from two LLM-as-a-Judge models (Factual Score, Bio Quality) described in III-H2. We also report the Overall Score for readability, defined as the mean of these two qualitative metrics. To ensure

TABLE 3. Overall results (8B Tier) (Mean ± Std). Metrics include perplexity (PPL) for fluency, ROUGE/BLEU for text overlap, 'Factual Score' (0-5, measuring accuracy of extracted attributes), 'Bio Quality' (0-5, assessing narrative coherence), and 'Overall Score' (average of the qualitative metrics).

Model	Rank	Dataset	PPL	ROUGE	BLEU	Factual Score	Bio Quality	Overall Score
DeepSeek-7B	0	base	11.600	0.134 ± 0.000	0.251 ± 0.004	2.385 ± 0.011	1.917 ± 0.016	2.151 ± 0.012
DeepSeek-7B	16	unstructured	6.450 ± 0.245	0.093 ± 0.003	0.647 ± 0.325	1.022 ± 0.414	1.879 ± 0.059	1.451 ± 0.235
DeepSeek-7B	32	unstructured	6.059 ± 0.154	0.105 ± 0.010	1.230 ± 0.351	1.794 ± 0.634	2.004 ± 0.073	1.899 ± 0.280
DeepSeek-7B	64	unstructured	5.703 ± 0.152	0.105 ± 0.016	1.465 ± 0.647	2.035 ± 0.922	1.916 ± 0.045	1.975 ± 0.484
DeepSeek-7B	16	structured	6.312 ± 0.174	0.118 ± 0.014	2.961 ± 0.884	2.974 ± 0.143	1.939 ± 0.165	2.457 ± 0.126
DeepSeek-7B	32	structured	5.898 ± 0.115	0.140 ± 0.007	4.319 ± 0.735	3.164 ± 0.048	1.865 ± 0.076	2.514 ± 0.028
DeepSeek-7B	64	structured	5.535 ± 0.295	0.153 ± 0.005	4.228 ± 1.719	3.090 ± 0.131	1.861 ± 0.132	2.476 ± 0.118
DeepSeek-7B	16	summarized	7.951 ± 0.176	0.239 ± 0.021	6.908 ± 1.490	3.005 ± 0.237	2.003 ± 0.161	2.504 ± 0.199
DeepSeek-7B	32	summarized	7.686 ± 0.079	0.244 ± 0.012	6.773 ± 1.024	2.999 ± 0.176	2.034 ± 0.068	2.517 ± 0.112
DeepSeek-7B	64	summarized	6.974 ± 0.278	0.249 ± 0.004	7.671 ± 0.118	2.956 ± 0.397	1.965 ± 0.113	2.460 ± 0.248
Llama-3.1-8B	0	base	10.947	0.100 ± 0.005	0.057 ± 0.049	1.677 ± 0.068	1.287 ± 0.011	1.482 ± 0.038
Llama-3.1-8B	16	unstructured	5.490 ± 0.178	0.102 ± 0.017	1.937 ± 0.363	2.674 ± 0.149	1.603 ± 0.068	2.139 ± 0.108
Llama-3.1-8B	32	unstructured	4.876 ± 0.257	0.108 ± 0.021	1.926 ± 0.437	2.546 ± 0.103	1.679 ± 0.041	2.113 ± 0.061
Llama-3.1-8B	64	unstructured	4.438 ± 0.193	0.104 ± 0.019	1.259 ± 0.521	2.616 ± 0.028	1.683 ± 0.031	2.149 ± 0.012
Llama-3.1-8B	16	structured	5.471 ± 0.133	0.123 ± 0.009	3.394 ± 0.339	2.800 ± 0.103	1.576 ± 0.049	2.188 ± 0.065
Llama-3.1-8B	32	structured	4.956 ± 0.241	0.118 ± 0.005	2.613 ± 0.275	2.901 ± 0.158	1.637 ± 0.046	2.269 ± 0.101
Llama-3.1-8B	64	structured	4.346 ± 0.197	0.137 ± 0.024	3.988 ± 1.739	2.898 ± 0.195	1.653 ± 0.127	2.276 ± 0.161
Llama-3.1-8B	16	summarized	6.857 ± 0.381	0.202 ± 0.016	4.482 ± 0.759	2.785 ± 0.379	1.671 ± 0.073	2.228 ± 0.224
Llama-3.1-8B	32	summarized	6.361 ± 0.125	0.138 ± 0.118	4.128 ± 3.711	2.981 ± 0.084	1.440 ± 0.379	2.210 ± 0.228
Llama-3.1-8B	64	summarized	5.903 ± 0.671	0.165 ± 0.131	4.657 ± 4.426	2.954 ± 0.250	1.791 ± 0.135	2.372 ± 0.191
Qwen3-8B	0	base	14.795	0.099 ± 0.000	0.080 ± 0.010	1.043 ± 0.009	1.537 ± 0.007	1.290 ± 0.002
Qwen3-8B	16	unstructured	6.853 ± 0.038	0.068 ± 0.036	0.558 ± 0.336	1.569 ± 0.780	1.420 ± 0.221	1.495 ± 0.494
Qwen3-8B	32	unstructured	6.530 ± 0.087	0.098 ± 0.035	1.138 ± 0.749	2.236 ± 0.605	1.609 ± 0.118	1.922 ± 0.361
Qwen3-8B	64	unstructured	5.872 ± 0.341	0.102 ± 0.030	1.553 ± 0.862	2.186 ± 0.807	1.588 ± 0.138	1.887 ± 0.470
Qwen3-8B	16	structured	6.892 ± 0.021	0.123 ± 0.004	2.493 ± 0.444	2.687 ± 0.294	1.657 ± 0.030	2.172 ± 0.152
Qwen3-8B	32	structured	6.579 ± 0.078	0.111 ± 0.011	1.996 ± 0.894	2.785 ± 0.136	1.512 ± 0.123	2.149 ± 0.117
Qwen3-8B	64	structured	5.846 ± 0.265	0.123 ± 0.004	3.209 ± 1.558	3.005 ± 0.105	1.661 ± 0.041	2.333 ± 0.043
Qwen3-8B	16	summarized	10.282 ± 0.449	0.208 ± 0.030	4.527 ± 1.579	2.820 ± 0.312	1.706 ± 0.168	2.263 ± 0.240
Qwen3-8B	32	summarized	9.451 ± 0.206	0.128 ± 0.040	1.907 ± 0.600	2.383 ± 0.290	1.418 ± 0.134	1.901 ± 0.181
Qwen3-8B	64	summarized	8.712 ± 0.228	0.199 ± 0.022	3.984 ± 1.661	2.787 ± 0.399	1.670 ± 0.170	2.229 ± 0.284

statistical robustness, every reported result was derived from three independent training and inference runs, initialized with different random seeds, shown in Table 2. For each run, the fine-tuned model generated responses for the test set of 269 characters. All metrics in the following tables are reported as Mean ± Standard Deviation (Std), providing a measure of both performance and training stability.

A. TIER-WISE PERFORMANCE ANALYSIS

The results are organized by model size tier to facilitate direct architectural comparisons. We first examine the ∼8B tier, followed by the ∼13B and ∼33B tiers, before aggregating the data to answer specific research questions regarding data formatting and LoRA rank.

1) OVERALL RESULTS: ∼8B TIER

Table 3 details the performance of the ∼8B parameter models. This tier offers the most direct architectural comparison, as all three models, DeepSeek-7B, Llama-3.1-8B, and Qwen3-8B, are modern, instruction-tuned iterations.

DeepSeek-7B exhibited remarkable zero-shot capabilities. Even without fine-tuning (Rank 0), it achieved a high Factual Score of 2.385 and a Bio Quality of 1.917, suggesting its pre-training corpus already contains significant knowledge of *The Elder Scrolls* lore. Fine-tuning further refined this capability; notably, training on the *structured* dataset with Rank 32 boosted the Factual Score to 3.164, the highest in this tier.

Llama-3.1-8B demonstrated exceptional adaptability. While its base model performance was lower than DeepSeek's, fine-tuning triggered drastic improvements. It achieved the lowest perplexity in the entire tier (4.346 with Rank 64/Structured), indicating superior fluency. Furthermore, it offered the most balanced performance profile; the Rank 64/Summarized configuration yielded a Factual Score of 2.954 while maintaining a competitive Overall Score of 2.372. This suggests the Llama architecture is particularly efficient at internalizing new stylistic and factual patterns simultaneously.

Qwen3-8B demonstrated the most dramatic relative improvement among all models in this tier. Starting with a baseline perplexity of 14.795, fine-tuning effectively bridged the domain gap, reducing perplexity to as low as 5.846 (Rank 64/Structured). This represents a substantial adaptation capability. Furthermore, its ability to generate structured factual data improved significantly, with the Factual Score nearly tripling from 1.043 (Base) to 3.005 (Rank 64/Structured). This trajectory indicates that while the base model may have had limited initial exposure to this specific lore, the architecture is highly responsive to QLoRA adaptation, rapidly internalizing the domain constraints.

Finally, a broader analysis of the quantitative metrics reveals a consistent correlation between adapter capacity and model performance in this tier. Across nearly all configurations, increasing the LoRA rank from 16 to 64 resulted in a monotonic decrease in Perplexity. For

TABLE 4. Overall results (13B Tier) (Mean ± Std). Metrics include Perplexity (PPL) for fluency, ROUGE/BLEU for text overlap, 'Factual Score' (0-5, measuring accuracy of extracted attributes), 'Bio Quality' (0-5, assessing narrative coherence), and 'Overall Score' (average of the qualitative metrics).

Model	Rank	Dataset	PPL	ROUGE	BLEU	Factual Score	Bio Quality	Overall Score
D-Coder-V2-16B	0	base	13.901	0.132 ± 0.001	0.298 ± 0.028	2.433 ± 0.004	2.193 ± 0.021	2.313 ± 0.012
D-Coder-V2-16B	16	unstructured	5.303 ± 0.020	0.124 ± 0.003	1.881 ± 0.204	2.907 ± 0.134	2.039 ± 0.065	2.473 ± 0.037
D-Coder-V2-16B	32	unstructured	4.260 ± 0.268	0.121 ± 0.004	1.651 ± 0.190	2.820 ± 0.234	2.167 ± 0.128	2.493 ± 0.181
D-Coder-V2-16B	64	unstructured	3.801 ± 0.015	0.116 ± 0.010	2.342 ± 0.642	2.907 ± 0.154	1.980 ± 0.058	2.444 ± 0.105
D-Coder-V2-16B	16	structured	4.662 ± 0.258	0.131 ± 0.008	4.089 ± 1.604	3.163 ± 0.096	2.152 ± 0.098	2.658 ± 0.081
D-Coder-V2-16B	32	structured	4.414 ± 0.118	0.140 ± 0.002	4.050 ± 0.862	3.295 ± 0.163	2.183 ± 0.069	2.739 ± 0.114
D-Coder-V2-16B	64	structured	3.820 ± 0.062	0.174 ± 0.027	4.181 ± 2.411	3.281 ± 0.095	2.147 ± 0.030	2.714 ± 0.058
D-Coder-V2-16B	16	summarized	7.149 ± 0.495	0.201 ± 0.013	5.523 ± 2.294	3.018 ± 0.258	2.115 ± 0.128	2.566 ± 0.193
D-Coder-V2-16B	32	summarized	6.830 ± 0.526	0.286 ± 0.078	14.767 ± 9.525	3.826 ± 0.244	2.292 ± 0.095	3.059 ± 0.167
D-Coder-V2-16B	64	summarized	6.052 ± 0.437	0.337 ± 0.063	20.872 ± 7.912	3.953 ± 0.335	2.372 ± 0.229	3.162 ± 0.279
Llama-2-13B	0	base	8.694	0.121 ± 0.001	0.124 ± 0.030	2.320 ± 0.005	1.689 ± 0.004	2.004 ± 0.004
Llama-2-13B	16	unstructured	4.754 ± 0.328	0.102 ± 0.009	0.750 ± 0.645	0.858 ± 0.089	1.476 ± 0.133	1.167 ± 0.106
Llama-2-13B	32	unstructured	4.376 ± 0.037	0.108 ± 0.002	1.674 ± 0.548	0.591 ± 0.346	1.568 ± 0.021	1.080 ± 0.183
Llama-2-13B	64	unstructured	3.911 ± 0.099	0.112 ± 0.006	1.398 ± 0.810	0.511 ± 0.069	1.580 ± 0.101	1.045 ± 0.084
Llama-2-13B	16	structured	4.503 ± 0.039	0.160 ± 0.010	2.580 ± 0.460	2.658 ± 0.175	1.575 ± 0.025	2.116 ± 0.099
Llama-2-13B	32	structured	4.314 ± 0.194	0.160 ± 0.008	4.118 ± 1.750	2.773 ± 0.121	1.665 ± 0.028	2.219 ± 0.071
Llama-2-13B	64	structured	3.940 ± 0.136	0.175 ± 0.002	4.600 ± 2.536	2.644 ± 0.176	1.716 ± 0.059	2.180 ± 0.080
Llama-2-13B	16	summarized	4.777 ± 0.254	0.257 ± 0.005	8.452 ± 1.033	2.842 ± 0.207	1.723 ± 0.008	2.282 ± 0.101
Llama-2-13B	32	summarized	4.228 ± 0.048	0.264 ± 0.009	9.462 ± 0.454	2.958 ± 0.216	1.748 ± 0.085	2.353 ± 0.150
Llama-2-13B	64	summarized	4.136 ± 0.182	0.275 ± 0.010	12.312 ± 1.174	3.264 ± 0.074	1.833 ± 0.033	2.549 ± 0.046
Qwen3-14B	0	base	12.758	0.098 ± 0.003	0.067 ± 0.020	1.265 ± 0.006	1.938 ± 0.013	1.601 ± 0.006
Qwen3-14B	16	unstructured	6.122 ± 0.145	0.125 ± 0.006	1.531 ± 0.341	2.555 ± 0.088	1.747 ± 0.062	2.151 ± 0.064
Qwen3-14B	32	unstructured	5.957 ± 0.101	0.129 ± 0.008	0.722 ± 0.573	2.724 ± 0.072	1.819 ± 0.114	2.272 ± 0.058
Qwen3-14B	64	unstructured	5.280 ± 0.225	0.116 ± 0.007	1.897 ± 0.186	2.718 ± 0.048	1.774 ± 0.263	2.246 ± 0.135
Qwen3-14B	16	structured	6.083 ± 0.072	0.118 ± 0.011	2.607 ± 0.327	2.705 ± 0.151	1.822 ± 0.131	2.264 ± 0.110
Qwen3-14B	32	structured	5.871 ± 0.080	0.141 ± 0.011	3.713 ± 0.771	2.851 ± 0.112	1.747 ± 0.064	2.299 ± 0.086
Qwen3-14B	64	structured	5.510 ± 0.152	0.129 ± 0.007	3.611 ± 0.575	2.578 ± 0.089	1.772 ± 0.059	2.175 ± 0.068
Qwen3-14B	16	summarized	8.035 ± 0.136	0.239 ± 0.026	6.777 ± 2.560	2.865 ± 0.425	1.865 ± 0.129	2.365 ± 0.270
Qwen3-14B	32	summarized	7.475 ± 0.220	0.219 ± 0.021	5.746 ± 2.167	2.664 ± 0.298	1.731 ± 0.081	2.197 ± 0.187
Qwen3-14B	64	summarized	7.254 ± 0.127	0.231 ± 0.054	7.671 ± 3.408	3.189 ± 0.279	1.859 ± 0.095	2.524 ± 0.185

example, Llama-3.1-8B's perplexity on the structured dataset improved from 5.471 (Rank 16) to 4.346 (Rank 64). This trend indicates that the additional trainable parameters allow the models to more effectively approximate the probability distribution of the target domain, leading to superior fluency. Concurrently, this increased capacity typically correlated with higher BLEU scores; notably, DeepSeek-7B's BLEU score on the structured dataset rose from 2.961 (Rank 16) to 4.228 (Rank 64). This suggests that higher-rank adapters enable a more precise alignment with the domain's specialized terminology and structural conventions, allowing the model to generate content that statistically resembles the ground truth without resorting to rote memorization.

2) OVERALL RESULTS: ∼13B TIER

Table 4 presents the results for the ∼13B parameter tier. It is important to recall that due to gaps in the current open-source landscape, specifically the absence of a 13B Llama 3 or a general-purpose DeepSeek model in this range, this tier necessitates comparing the older Llama 2 architecture against the code-specialized DeepSeek-V2 (16B) and the modern Qwen3. Unlike the previous tier where the general-purpose Llama model led, this unique cross-generational comparison reveals curious results for specialized architectures.

DeepSeek-Coder-V2-16B emerged as the dominant model, driven by exceptional baseline capabilities. Even at Rank 0, it achieved a high Factual Score of 2.433 and a Bio Quality of 2.193, suggesting that its pre-training on code, inherently rich in logic and structure, transfers effectively to the structured lore extraction task. Fine-tuning unlocked substantial gains, particularly on the summarized dataset with Rank 64, where it reached a remarkable Factual Score of 3.953 and an Overall Score of 3.162. While explicitly marketed as a "Coder" model, DeepSeek's strong performance suggests its pre-training corpus likely included a substantial volume of diverse, high-quality general text alongside code.

Llama-2-13B, representing the previous generation of dense models, exhibited a different performance profile. Similar to DeepSeek, it started with a strong baseline Factual Score of 2.320, significantly higher than Qwen's. However, the results underscore the formidable challenge of overcoming a superior pre-training baseline. Notably, the Llama model was only able to surpass the Overall Score of the untrained (Rank 0) DeepSeek model (2.313) when fine-tuned with higher ranks (32 and 64) on the summarized dataset. This indicates that while fine-tuning can significantly improve performance, the initial depth of domain knowledge embedded during pre-training remains a decisive factor that is difficult to fully offset with parameter-efficient adaptations alone.

Qwen3-14B followed a trajectory similar to its 8B counterpart, starting with a lower baseline (Factual Score 1.265) but demonstrating robust adaptation. Fine-tuning effectively brought its performance into parity with the Llama-2 model

(Overall Score ≈ 2.52). However, it remained unable to bridge the gap to the DeepSeek Coder, reinforcing the observation that in this specific mid-sized tier, architectural specialization provided a greater advantage than general-purpose scaling.

3) OVERALL RESULTS: ∼33B TIER

Table 5 details the performance of the largest models in our study. This tier is particularly notable for testing whether the "code-specialized" hypothesis holds at scale, comparing heavyweights like CodeLlama and DeepSeek-Coder against the generalist Qwen3.

DeepSeek-Coder-33B provided the most distinct behavioral profile in the study. Unlike its 16B sibling, this model exhibited the characteristics of a "true" specialized coder. Its base performance was near zero (Overall Score 0.325), indicating negligible prior knowledge of the lore. However, fine-tuning catalyzed a massive transformation: on the summarized dataset (Rank 64), its Factual Score skyrocketed to 3.317. This confirms that the QLoRA pipeline successfully injected the domain knowledge from scratch, rather than merely realigning existing representations, validating the efficacy of the training methodology.

CodeLlama-34B mirrored this trend but benefited from a stronger starting position (Overall Score 1.266), likely inheriting some general knowledge from its Llama 2 foundation. Post-fine-tuning, it achieved the highest Factual Score in the tier (3.347 on Rank 64/Summarized). This reinforces the adaptability of the Llama architecture; even when heavily pre-trained on code, it retains a plasticity that allows for exceptional performance in structured narrative tasks.

Qwen3-32B presented a contrasting baseline, starting with significantly higher general capability (Overall Score 1.838) than the coder models. While it achieved robust fine-tuned performance, particularly in fluency (PPL 4.888), it did not exhibit the same magnitude of relative improvement in factual extraction as the specialized models.

B. MODEL SCALE RESULTS (EXPERIMENT 1)

To assess the effect of model size on fine-tuning performance, the results were averaged across all dataset types and fine-tuned LoRA ranks (16, 32, 64) for each model, as shown in Table 6. By excluding the base (Rank 0) models, this analysis attempts to isolate the adaptability of the architectures.

A nuanced trend emerges regarding model scale. Within the Qwen and Llama families, increasing parameter counts generally correlated with improved fluency. For instance, the Qwen series exhibited a monotonic decrease in Perplexity (PPL) from 7.520 (8B) to 6.225 (32B), and the Llama family improved from 5.415 (8B) to ≈4.3 (13B/34B). However, the results also highlight that extreme domain specialization can negate the advantages of scale. A prime example is the DeepSeek family: the modern DeepSeek-Coder-V2-16B significantly outperformed the larger DeepSeek-Coder-33B in both fluency (PPL 5.160 vs. 5.667) and Factual Score (3.246 vs. 2.751). This inversion is likely driven by the 33B model's narrower pre-training focus on code; as observed in the base model evaluation, it possessed negligible prior lore knowledge compared to the 16B variant. Consequently, despite its larger capacity, the 33B model was forced to learn the domain from scratch, whereas the newer 16B architecture benefited from a richer, more generalist foundation.

Furthermore, the intermediate ∼13B tier emerged as a "sweet spot" for qualitative performance, with the DeepSeek-V2-16B achieving the highest Overall Score in the entire study (2.704), surpassing all 33B models. However, the ∼8B tier also demonstrated impressive capability, validating the viability of consumer-grade fine-tuning. Notably, the Llama-3.1-8B model achieved a Factual Score (2.796) that is competitive with the much larger CodeLlama-34B (2.602). These findings indicate that for domain-specific fine-tuning, selecting a modern, efficient architecture often yields better results than simply scaling up to the largest available model.

C. DATA FORMATTING RESULTS (EXPERIMENT 2)

The structure of the fine-tuning data had a profound and often counter-intuitive impact on model performance, as detailed in Table 7. As hypothesized, fine-tuning on the structured dataset consistently resulted in the lowest perplexity scores across almost all models, indicating the highest level of stylistic and linguistic adaptation. For instance, the DeepSeek-Coder-V2-16B model saw its Perplexity improve from 4.450 (Unstructured) to 4.302 (Structured), confirming that the key-value pairing format aids in learning the domain's syntax.

The most striking finding, however, is the "density paradox" observed with the summarized dataset. Despite consistently producing the worst fluency metrics, exemplified by Qwen3-8B's perplexity spiking to 9.485, this high-density format yielded the highest Factual and Bio Quality scores for the majority of the mid-to-large scale models. The CodeLlama-34B model provides the definitive example of this phenomenon: while its Structured Factual Score was a modest 2.254, training on the Summarized dataset propelled it to 3.352, the highest in the entire experiment. Similarly, Llama-2-13B saw its factual accuracy rise from 2.694 (Structured) to 3.022 (Summarized).

However, this high-density strategy proved to be a high-variance variable, dependent on the intrinsic robustness of the architecture rather than scale alone. While the Llama family, including the small Llama-3.1-8B, effectively used the summarized data to maintain or improve factual scoring, other models struggled. Specifically, Qwen3-8B and DeepSeek-7B saw their Factual Scores regress on the summarized dataset compared to the structured baseline. This points to a clear practical conclusion: structured data acts as a "safety net," ensuring consistent performance improvements across all architectures, whereas summarized data represents a high-risk, high-reward approach that requires a compatible base model to effectively unpack the dense factual relationships.

TABLE 5. Overall results (33B Tier) (Mean ± Std). Metrics include perplexity (PPL) for fluency, ROUGE/BLEU for text overlap, 'Factual Score' (0-5, measuring accuracy of extracted attributes), 'Bio Quality' (0-5, assessing narrative coherence), and 'Overall Score' (average of the qualitative metrics).

Model	Rank	Dataset	PPL	ROUGE	BLEU	Factual Score	Bio Quality	Overall Score
D-Coder-33B	0	base	12.460	0.071 ± 0.001	0.024 ± 0.007	0.078 ± 0.017	0.572 ± 0.008	0.325 ± 0.005
D-Coder-33B	16	unstructured	6.310 ± 0.186	0.105 ± 0.007	1.564 ± 0.126	2.156 ± 0.500	1.848 ± 0.187	2.002 ± 0.322
D-Coder-33B	32	unstructured	5.884 ± 0.197	0.113 ± 0.008	1.387 ± 0.524	2.380 ± 0.219	1.667 ± 0.035	2.023 ± 0.124
D-Coder-33B	64	unstructured	5.321 ± 0.321	0.099 ± 0.015	1.564 ± 0.247	2.092 ± 0.781	1.748 ± 0.129	1.920 ± 0.433
D-Coder-33B	16	structured	6.221 ± 0.453	0.126 ± 0.012	3.535 ± 0.681	2.690 ± 0.178	1.742 ± 0.065	2.216 ± 0.120
D-Coder-33B	32	structured	5.908 ± 0.245	0.110 ± 0.028	2.358 ± 1.989	2.790 ± 0.127	1.696 ± 0.188	2.243 ± 0.133
D-Coder-33B	64	structured	5.267 ± 0.088	0.111 ± 0.019	1.619 ± 1.416	3.216 ± 0.076	1.616 ± 0.255	2.416 ± 0.163
D-Coder-33B	16	summarized	6.355 ± 0.568	0.224 ± 0.030	6.142 ± 1.807	2.869 ± 0.220	1.815 ± 0.045	2.342 ± 0.132
D-Coder-33B	32	summarized	5.137 ± 0.369	0.212 ± 0.022	5.593 ± 0.740	3.122 ± 0.076	1.853 ± 0.009	2.488 ± 0.039
D-Coder-33B	64	summarized	4.665 ± 0.060	0.237 ± 0.015	7.432 ± 1.301	3.317 ± 0.111	1.873 ± 0.025	2.595 ± 0.065
CodeLlama-34B	0	base	8.573	0.135 ± 0.000	0.999 ± 0.011	1.076 ± 0.019	1.456 ± 0.017	1.266 ± 0.016
CodeLlama-34B	16	unstructured	4.692 ± 0.109	0.126 ± 0.004	1.033 ± 0.106	2.137 ± 0.063	1.727 ± 0.073	1.932 ± 0.068
CodeLlama-34B	32	unstructured	4.309 ± 0.086	0.122 ± 0.004	1.925 ± 0.210	2.270 ± 0.063	1.649 ± 0.054	1.959 ± 0.057
CodeLlama-34B	64	unstructured	3.707 ± 0.243	0.116 ± 0.005	1.669 ± 0.383	2.169 ± 0.169	1.632 ± 0.076	1.900 ± 0.059
CodeLlama-34B	16	structured	4.617 ± 0.229	0.133 ± 0.012	3.314 ± 0.493	2.145 ± 0.532	1.575 ± 0.052	1.860 ± 0.289
CodeLlama-34B	32	structured	4.272 ± 0.165	0.129 ± 0.032	2.309 ± 1.884	2.337 ± 0.275	1.572 ± 0.141	1.955 ± 0.125
CodeLlama-34B	64	structured	3.889 ± 0.246	0.122 ± 0.044	3.925 ± 1.736	2.265 ± 0.503	1.614 ± 0.066	1.940 ± 0.261
CodeLlama-34B	16	summarized	4.728 ± 0.159	0.281 ± 0.020	13.510 ± 3.289	3.340 ± 0.226	1.828 ± 0.020	2.584 ± 0.122
CodeLlama-34B	32	summarized	4.468 ± 0.174	0.275 ± 0.023	11.667 ± 1.938	3.361 ± 0.217	1.781 ± 0.023	2.571 ± 0.114
CodeLlama-34B	64	summarized	4.103 ± 0.087	0.307 ± 0.059	16.339 ± 7.443	3.347 ± 0.277	1.954 ± 0.138	2.651 ± 0.183
Qwen3-32B	0	base	12.071	0.105 ± 0.001	0.078 ± 0.026	1.571 ± 0.025	2.104 ± 0.006	1.838 ± 0.014
Qwen3-32B	16	unstructured	5.958 ± 0.283	0.126 ± 0.005	1.205 ± 0.413	2.739 ± 0.116	1.808 ± 0.081	2.273 ± 0.096
Qwen3-32B	32	unstructured	5.453 ± 0.194	0.126 ± 0.002	1.682 ± 0.601	2.709 ± 0.039	1.876 ± 0.008	2.292 ± 0.023
Qwen3-32B	64	unstructured	4.954 ± 0.305	0.126 ± 0.002	1.368 ± 0.715	2.854 ± 0.183	1.868 ± 0.134	2.361 ± 0.159
Qwen3-32B	16	structured	5.804 ± 0.107	0.118 ± 0.005	2.223 ± 0.185	2.944 ± 0.116	1.943 ± 0.024	2.443 ± 0.069
Qwen3-32B	32	structured	5.569 ± 0.153	0.112 ± 0.017	1.729 ± 0.816	3.073 ± 0.199	1.904 ± 0.103	2.489 ± 0.064
Qwen3-32B	64	structured	4.888 ± 0.191	0.119 ± 0.003	2.964 ± 0.542	3.087 ± 0.067	2.048 ± 0.103	2.567 ± 0.081
Qwen3-32B	16	summarized	8.226 ± 0.601	0.180 ± 0.074	4.100 ± 2.551	2.914 ± 0.476	1.607 ± 0.441	2.261 ± 0.458
Qwen3-32B	32	summarized	7.396 ± 0.056	0.185 ± 0.012	3.710 ± 0.542	2.891 ± 0.386	1.777 ± 0.214	2.334 ± 0.296
Qwen3-32B	64	summarized	7.497 ± 0.283	0.205 ± 0.027	4.805 ± 1.687	2.935 ± 0.159	1.730 ± 0.076	2.333 ± 0.115

TABLE 6. Experiment 1: Impact of model scale (Averaged over ranks > 0 and datasets). Metrics include perplexity (PPL) for fluency, ROUGE/BLEU for text overlap, 'Factual Score' (0-5, measuring accuracy of extracted attributes), 'Bio Quality' (0-5, assessing narrative coherence), and 'Overall Score' (average of the qualitative metrics).

Model	PPL	ROUGE	BLEU	Factual Score	Bio Quality	Overall Score
DeepSeek-7B	6.536 ± 0.110	0.165 ± 0.003	4.240 ± 0.198	2.648 ± 0.089	1.944 ± 0.037	2.296 ± 0.038
Llama-3.1-8B	5.415 ± 0.169	0.133 ± 0.027	3.168 ± 0.735	2.796 ± 0.068	1.637 ± 0.034	2.217 ± 0.049
Qwen3-8B	7.520 ± 0.158	0.133 ± 0.009	2.488 ± 0.179	2.557 ± 0.094	1.593 ± 0.037	2.075 ± 0.066
D-Coder-V2-16B	5.160 ± 0.035	0.182 ± 0.012	6.644 ± 1.659	3.246 ± 0.068	2.162 ± 0.061	2.704 ± 0.063
Llama-2-13B	4.331 ± 0.050	0.184 ± 0.005	5.276 ± 0.210	2.209 ± 0.061	1.660 ± 0.020	1.935 ± 0.036
Qwen3-14B	6.418 ± 0.034	0.162 ± 0.013	3.865 ± 0.850	2.763 ± 0.100	1.794 ± 0.042	2.278 ± 0.071
D-Coder-33B	5.667 ± 0.053	0.152 ± 0.011	3.590 ± 0.639	2.751 ± 0.133	1.767 ± 0.006	2.259 ± 0.069
CodeLlama-34B	4.312 ± 0.056	0.179 ± 0.003	6.211 ± 0.476	2.602 ± 0.126	1.705 ± 0.025	2.154 ± 0.073
Qwen3-32B	6.225 ± 0.050	0.145 ± 0.006	2.688 ± 0.382	2.905 ± 0.039	1.838 ± 0.026	2.372 ± 0.007

D. LoRA RANK COMPARISON RESULTS (EXPERIMENT 3)

The analysis of LoRA rank reveals a nuanced relationship between adapter capacity and performance, as shown in Table 8. For the majority of general-purpose architectures, increasing the rank from 16 to 64 resulted in consistent, monotonic improvements. The Llama-3.1-8B model exemplifies this trend: its average Perplexity improved significantly from 5.948 (Rank 16) to 4.896 (Rank 64), while its Overall Score rose from 2.186 to 2.267. Similarly, the Qwen3 and DeepSeek-7B (base) models showed clear gains in both fluency and qualitative scoring with the larger adapter, indicating that for standard dense transformers, additional trainable parameters are necessary to fully capture the domain's complexities.

However, the code-specialized models exhibited a distinct behavior. For instance, while DeepSeek-Coder-V2-16B showed a strong jump from Rank 16 to 32, increasing further to Rank 64 yielded diminishing returns, with the Bio Quality score actually regressing slightly (2.214 to 2.168) despite the lower perplexity. A similar pattern was observed with CodeLlama-34B, where the Overall Score plateaued completely at 2.165 between Rank 32 and 64. This suggests that for models already heavily pre-trained on structured data, a moderate rank of 32 may represent an optimal efficiency point, providing sufficient plasticity for adaptation without requiring larger adaptation matrices.

E. ARCHITECTURAL COMPARISON RESULTS (EXPERIMENT 4)

The next analysis, based on the win rates determined by the ensemble LLM-as-a-Judge, provides a clear qualitative comparison of the foundational architectures within each

TABLE 7. Experiment 2: Efficacy of data formatting (Averaged over ranks > 0). Metrics include perplexity (PPL) for fluency, ROUGE/BLEU for text overlap, 'Factual Score' (0-5, measuring accuracy of extracted attributes), 'Bio Quality' (0-5, assessing narrative coherence), and 'Overall Score' (average of the qualitative metrics).

Model	Dataset	PPL	ROUGE	BLEU	Factual Score	Bio Quality	Overall Score
DeepSeek-7B	unstructured	6.044 ± 0.100	0.101 ± 0.009	1.146 ± 0.157	1.652 ± 0.599	1.938 ± 0.019	1.795 ± 0.303
DeepSeek-7B	structured	5.915 ± 0.163	0.137 ± 0.007	3.828 ± 0.977	3.077 ± 0.019	1.888 ± 0.042	2.482 ± 0.013
DeepSeek-7B	summarized	7.544 ± 0.109	0.244 ± 0.012	7.107 ± 0.823	2.988 ± 0.226	2.001 ± 0.101	2.495 ± 0.163
Llama-3.1-8B	unstructured	4.933 ± 0.181	0.105 ± 0.015	1.708 ± 0.124	2.612 ± 0.016	1.655 ± 0.030	2.134 ± 0.021
Llama-3.1-8B	structured	4.923 ± 0.166	0.126 ± 0.005	3.346 ± 0.431	2.867 ± 0.022	1.623 ± 0.041	2.245 ± 0.022
Llama-3.1-8B	summarized	6.378 ± 0.238	0.169 ± 0.080	4.435 ± 2.498	2.906 ± 0.170	1.634 ± 0.156	2.270 ± 0.152
Qwen3-8B	unstructured	6.391 ± 0.126	0.094 ± 0.013	1.170 ± 0.483	2.099 ± 0.278	1.564 ± 0.041	1.831 ± 0.154
Qwen3-8B	structured	6.444 ± 0.115	0.119 ± 0.003	2.562 ± 0.905	2.823 ± 0.065	1.610 ± 0.040	2.217 ± 0.043
Qwen3-8B	summarized	9.485 ± 0.176	0.179 ± 0.021	3.483 ± 0.797	2.666 ± 0.309	1.599 ± 0.106	2.133 ± 0.206
D-Coder-V2-16B	unstructured	4.450 ± 0.091	0.121 ± 0.004	1.951 ± 0.107	2.879 ± 0.081	2.063 ± 0.030	2.471 ± 0.045
D-Coder-V2-16B	structured	4.302 ± 0.054	0.148 ± 0.009	4.106 ± 0.042	3.248 ± 0.065	2.161 ± 0.062	2.704 ± 0.064
D-Coder-V2-16B	summarized	6.677 ± 0.235	0.274 ± 0.042	13.671 ± 4.929	3.598 ± 0.234	2.259 ± 0.137	2.928 ± 0.185
Llama-2-13B	unstructured	4.363 ± 0.111	0.107 ± 0.004	1.271 ± 0.452	0.658 ± 0.103	1.540 ± 0.018	1.099 ± 0.058
Llama-2-13B	structured	4.252 ± 0.098	0.165 ± 0.005	3.764 ± 0.752	2.694 ± 0.075	1.652 ± 0.016	2.173 ± 0.039
Llama-2-13B	summarized	4.383 ± 0.142	0.265 ± 0.007	10.081 ± 0.697	3.022 ± 0.125	1.768 ± 0.038	2.395 ± 0.072
Qwen3-14B	unstructured	5.788 ± 0.155	0.124 ± 0.004	1.378 ± 0.086	2.668 ± 0.055	1.781 ± 0.092	2.224 ± 0.071
Qwen3-14B	structured	5.817 ± 0.053	0.129 ± 0.007	3.313 ± 0.301	2.710 ± 0.112	1.780 ± 0.025	2.245 ± 0.067
Qwen3-14B	summarized	7.597 ± 0.158	0.230 ± 0.029	6.715 ± 2.225	2.900 ± 0.264	1.818 ± 0.073	2.359 ± 0.168
D-Coder-33B	unstructured	5.833 ± 0.125	0.106 ± 0.007	1.508 ± 0.252	2.212 ± 0.306	1.756 ± 0.099	1.984 ± 0.182
D-Coder-33B	structured	5.803 ± 0.084	0.116 ± 0.012	2.554 ± 0.574	2.896 ± 0.041	1.688 ± 0.090	2.292 ± 0.065
D-Coder-33B	summarized	5.395 ± 0.116	0.224 ± 0.021	6.385 ± 1.181	3.100 ± 0.123	1.847 ± 0.016	2.474 ± 0.069
CodeLlama-34B	unstructured	4.234 ± 0.074	0.121 ± 0.001	1.545 ± 0.163	2.193 ± 0.093	1.669 ± 0.016	1.931 ± 0.053
CodeLlama-34B	structured	4.265 ± 0.177	0.128 ± 0.008	3.200 ± 0.660	2.254 ± 0.228	1.588 ± 0.041	1.921 ± 0.134
CodeLlama-34B	summarized	4.434 ± 0.125	0.288 ± 0.008	13.848 ± 1.485	3.352 ± 0.120	1.855 ± 0.035	2.603 ± 0.048
Qwen3-32B	unstructured	5.442 ± 0.257	0.126 ± 0.001	1.420 ± 0.469	2.768 ± 0.039	1.851 ± 0.035	2.309 ± 0.036
Qwen3-32B	structured	5.419 ± 0.069	0.117 ± 0.005	2.320 ± 0.409	3.033 ± 0.079	1.967 ± 0.041	2.500 ± 0.041
Qwen3-32B	summarized	7.704 ± 0.091	0.190 ± 0.017	4.216 ± 0.480	2.916 ± 0.056	1.707 ± 0.054	2.311 ± 0.023

size tier, as visualized in Fig. 3. The results establish the Llama family as the most consistent performer for narrative generation tasks. It achieved the dominant win rate in both the ∼8B tier (48.3%) and the ∼33B tier (40.0%), outperforming its peers by a significant margin. This suggests that the Llama architecture possesses a robust general-purpose foundation that adapts exceptionally well to the stylistic nuances of lore, regardless of the specific fine-tuning scale.

In contrast, the ∼13B tier highlighted the specific strengths of the DeepSeek architecture. Here, the DeepSeek-Coder-V2 (16B) achieved a remarkable win rate (46.0%), decisively beating the Llama-2 (33.1%). This reinforces the finding that the logical reasoning capabilities inherent in the modern DeepSeek V2 architecture can be effectively transferred to structured knowledge extraction, allowing it to punch above its weight class. Finally, the Qwen family exhibited a clear scaling dependency; while it struggled significantly in the 8B tier (16.5%), it recovered competitiveness in the 33B tier (31.7%), effectively tying with or surpassing the older DeepSeek-33B model, though still trailing the CodeLlama variant.

F. CATASTROPHIC FORGETTING ANALYSIS (EXPERIMENT 5)

The impact of fine-tuning on general knowledge retention is detailed in Table 9. The results isolate a critical vulnerability in specific architectures: while Llama-3.1-8B demonstrated exceptional resilience, maintaining ≈62% accuracy across all configurations, the Llama-2 and Qwen series suffered significant degradation. This effect was most pronounced on the Summarized dataset; for instance, Llama-2-13B's accuracy plummeted from 61.7% (base) to 27.0% when fine-tuned with Rank 64 on summarized data. Conversely, the unstructured dataset, despite being more verbose, preserved general knowledge effectively across almost all models.

The data also reveals a direct correlation between adapter capacity and forgetting. As observed in the Llama-2-13B and Qwen3-32B trials, increasing the LoRA rank from 32 to 64 consistently exacerbated the performance drop on the summarized dataset. This indicates that while higher ranks provide the capacity needed to learn dense lore (as seen in Experiment 2), they simultaneously provide the "freedom" to overwrite the pre-trained weights responsible for general knowledge. Finally, the DeepSeek-33B model exhibited anomalously low performance on this specific benchmark (≈8%), which is an artifact of its code-centric pre-training interfering with the specific zero-shot trivia prompt format.

G. QUALITATIVE ERROR ANALYSIS

To illustrate the specific challenges of adapting these architectures, Table 10 presents representative outputs that highlight common failure modes encountered during the study. The analysis identifies three primary error categories:

1) DOMAIN ALIGNMENT REFUSAL

The base DeepSeek-Coder-33B model (Rank 0) provides a clear example of rigid pre-training alignment. When asked about *Arcadia*, it refused to answer, explicitly stating that

TABLE 8. Experiment 3: LoRA rank analysis (Averaged over datasets). Metrics include perplexity (PPL) for fluency, ROUGE/BLEU for text overlap, 'Factual Score' (0-5, measuring accuracy of extracted attributes), 'Bio Quality' (0-5, assessing narrative coherence), and 'Overall Score' (average of the qualitative metrics).

Model	Rank	PPL	ROUGE	BLEU	Factual Score	Bio Quality	Overall Score
DeepSeek-7B	0	11.600	0.134 ± 0.000	0.251 ± 0.004	2.385 ± 0.011	1.917 ± 0.016	2.151 ± 0.012
DeepSeek-7B	16	6.952 ± 0.103	0.157 ± 0.004	3.815 ± 0.242	2.483 ± 0.085	1.951 ± 0.096	2.217 ± 0.088
DeepSeek-7B	32	6.584 ± 0.011	0.167 ± 0.005	4.285 ± 0.260	2.708 ± 0.095	1.965 ± 0.010	2.336 ± 0.047
DeepSeek-7B	64	6.089 ± 0.259	0.172 ± 0.002	4.600 ± 0.626	2.747 ± 0.166	1.916 ± 0.048	2.332 ± 0.067
Llama-3.1-8B	0	10.947	0.100 ± 0.005	0.057 ± 0.049	1.677 ± 0.068	1.287 ± 0.011	1.482 ± 0.038
Llama-3.1-8B	16	5.948 ± 0.094	0.143 ± 0.008	3.288 ± 0.240	2.755 ± 0.134	1.617 ± 0.036	2.186 ± 0.079
Llama-3.1-8B	32	5.400 ± 0.122	0.121 ± 0.037	2.890 ± 1.147	2.809 ± 0.095	1.585 ± 0.137	2.197 ± 0.116
Llama-3.1-8B	64	4.896 ± 0.343	0.136 ± 0.038	3.325 ± 1.060	2.825 ± 0.031	1.710 ± 0.021	2.267 ± 0.020
Qwen3-8B	0	14.795	0.099 ± 0.000	0.080 ± 0.010	1.043 ± 0.009	1.537 ± 0.007	1.290 ± 0.002
Qwen3-8B	16	8.176 ± 0.213	0.142 ± 0.019	2.794 ± 0.691	2.473 ± 0.325	1.620 ± 0.108	2.047 ± 0.217
Qwen3-8B	32	7.551 ± 0.148	0.113 ± 0.020	1.700 ± 0.593	2.489 ± 0.152	1.511 ± 0.114	2.000 ± 0.132
Qwen3-8B	64	6.873 ± 0.184	0.144 ± 0.003	2.991 ± 0.478	2.694 ± 0.107	1.646 ± 0.036	2.170 ± 0.037
D-Coder-V2-16B	0	13.901	0.132 ± 0.001	0.298 ± 0.028	2.433 ± 0.004	2.193 ± 0.021	2.313 ± 0.012
D-Coder-V2-16B	16	5.724 ± 0.109	0.153 ± 0.008	3.872 ± 1.351	3.032 ± 0.067	2.103 ± 0.097	2.568 ± 0.082
D-Coder-V2-16B	32	5.188 ± 0.121	0.183 ± 0.027	6.877 ± 2.990	3.320 ± 0.127	2.214 ± 0.047	2.767 ± 0.087
D-Coder-V2-16B	64	4.571 ± 0.129	0.210 ± 0.020	9.211 ± 3.427	3.385 ± 0.039	2.168 ± 0.060	2.776 ± 0.049
Llama-2-13B	0	8.694	0.121 ± 0.001	0.124 ± 0.030	2.320 ± 0.005	1.689 ± 0.004	2.004 ± 0.004
Llama-2-13B	16	4.676 ± 0.056	0.177 ± 0.006	4.139 ± 0.296	2.181 ± 0.116	1.598 ± 0.034	1.890 ± 0.045
Llama-2-13B	32	4.304 ± 0.074	0.180 ± 0.005	5.204 ± 0.642	2.172 ± 0.193	1.664 ± 0.044	1.918 ± 0.117
Llama-2-13B	64	4.004 ± 0.135	0.194 ± 0.006	6.528 ± 1.148	2.273 ± 0.073	1.721 ± 0.006	1.997 ± 0.038
Qwen3-14B	0	12.758	0.098 ± 0.003	0.067 ± 0.020	1.265 ± 0.006	1.938 ± 0.013	1.601 ± 0.006
Qwen3-14B	16	6.789 ± 0.025	0.163 ± 0.012	3.752 ± 1.054	2.716 ± 0.170	1.815 ± 0.040	2.266 ± 0.086
Qwen3-14B	32	6.448 ± 0.038	0.164 ± 0.012	3.424 ± 0.991	2.744 ± 0.067	1.765 ± 0.030	2.255 ± 0.019
Qwen3-14B	64	6.014 ± 0.064	0.159 ± 0.023	4.424 ± 1.396	2.827 ± 0.114	1.802 ± 0.137	2.315 ± 0.124
D-Coder-33B	0	12.460	0.071 ± 0.001	0.024 ± 0.007	0.078 ± 0.017	0.572 ± 0.008	0.325 ± 0.005
D-Coder-33B	16	6.295 ± 0.052	0.154 ± 0.011	3.854 ± 0.592	2.590 ± 0.179	1.802 ± 0.042	2.196 ± 0.099
D-Coder-33B	32	5.619 ± 0.126	0.148 ± 0.014	3.209 ± 1.004	2.774 ± 0.093	1.743 ± 0.052	2.259 ± 0.053
D-Coder-33B	64	5.070 ± 0.094	0.152 ± 0.010	3.685 ± 0.388	2.896 ± 0.253	1.752 ± 0.043	2.324 ± 0.123
CodeLlama-34B	0	8.573	0.135 ± 0.000	0.999 ± 0.011	1.076 ± 0.019	1.456 ± 0.017	1.266 ± 0.016
CodeLlama-34B	16	4.680 ± 0.100	0.180 ± 0.008	5.977 ± 0.991	2.546 ± 0.212	1.712 ± 0.036	2.129 ± 0.121
CodeLlama-34B	32	4.351 ± 0.036	0.176 ± 0.016	5.345 ± 1.196	2.661 ± 0.152	1.669 ± 0.069	2.165 ± 0.080
CodeLlama-34B	64	3.900 ± 0.109	0.182 ± 0.032	7.287 ± 3.033	2.595 ± 0.287	1.735 ± 0.065	2.165 ± 0.146
Qwen3-32B	0	12.071	0.105 ± 0.001	0.078 ± 0.026	1.571 ± 0.025	2.104 ± 0.006	1.838 ± 0.014
Qwen3-32B	16	6.706 ± 0.162	0.142 ± 0.026	2.565 ± 1.029	2.869 ± 0.181	1.782 ± 0.182	2.325 ± 0.181
Qwen3-32B	32	6.175 ± 0.104	0.143 ± 0.009	2.418 ± 0.558	2.887 ± 0.205	1.852 ± 0.045	2.370 ± 0.123
Qwen3-32B	64	5.804 ± 0.212	0.151 ± 0.010	3.074 ± 0.614	2.958 ± 0.095	1.879 ± 0.059	2.418 ± 0.074

the query did not relate to "computer science" or "coding problems." This confirms that for specialized models, QLoRA fine-tuning is not just about injecting knowledge, but also about breaking the semantic guardrails established during pre-training.

2) HALLUCINATION OF NON-EXISTENCE

Both Qwen base models (8B and 14B) exhibited a tendency to confidently hallucinate that valid characters did not exist. As seen with *Salma*, the model not only failed to retrieve details but also actively generated a denial ("Salma is not a recognized character"), attributing the name to mods or fan fiction. This highlights a specific failure mode in the Qwen base models where uncertainty is resolved as negation rather than ambiguity.

3) BIOGRAPHICAL GENERICIZATION

The fine-tuned DeepSeek-V2-16B (Rank 64, Summarized) demonstrates a subtle but critical failure mode with *Iddra*. While the model flawlessly retrieves her attributes (Nord, Innkeeper, Kynesgrove), it fails to retrieve her specific narrative (her suspicion of her husband's infidelity). Instead, it hallucinates a generic backstory common to the domain ("frustrated by the war"). This indicates that the model successfully learned the entity's existence but filled the narrative gaps with statistically probable tropes rather than specific ground truth.

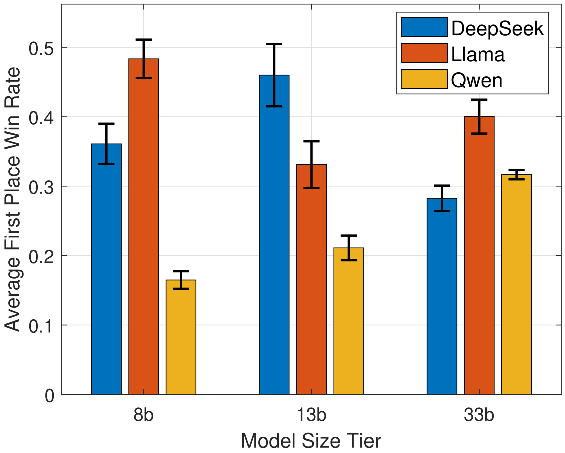


FIGURE 3. Experiment 4: Cross-Architectural Win Rate by Size Tier (Averaged over all configurations). Win rate represents the frequency with which a model's response was ranked as 'Best' by the ensemble judges (Gemini + EVA-Qwen) based on factual accuracy and narrative quality.

TABLE 9. Experiment 5: Catastrophic forgetting evaluation (Mean ± Std).

Model Name	Dataset	Base Accuracy (%)	Rank 16 Accuracy (%)	Rank 32 Accuracy (%)	Rank 64 Accuracy (%)
DeepSeek-7B	unstructured	48.3 ± 3.5	45.3 ± 7.5	45.3 ± 10.5	42.7 ± 7.8
DeepSeek-7B	structured	48.3 ± 3.5	29.7 ± 4.7	31.0 ± 8.7	35.7 ± 3.2
DeepSeek-7B	summarized	48.3 ± 3.5	32.0 ± 2.6	34.0 ± 6.0	35.0 ± 7.8
Llama-3.1-8B	unstructured	65.7 ± 1.5	62.7 ± 1.5	63.0 ± 1.0	61.7 ± 2.9
Llama-3.1-8B	structured	65.7 ± 1.5	62.7 ± 3.2	57.7 ± 2.9	61.3 ± 2.5
Llama-3.1-8B	summarized	65.7 ± 1.5	62.3 ± 1.2	62.7 ± 1.2	62.7 ± 3.1
Qwen3-8B	unstructured	71.3 ± 4.5	41.3 ± 4.6	47.0 ± 8.2	49.0 ± 5.3
Qwen3-8B	structured	71.3 ± 4.5	28.3 ± 7.1	49.7 ± 5.1	48.3 ± 8.3
Qwen3-8B	summarized	71.3 ± 4.5	38.7 ± 20.8	38.0 ± 14.4	52.0 ± 6.1
D-Coder-V2-16B	unstructured	57.3 ± 1.5	51.3 ± 7.1	48.3 ± 1.2	49.3 ± 5.0
D-Coder-V2-16B	structured	57.3 ± 1.5	54.3 ± 5.8	50.7 ± 7.2	49.7 ± 6.5
D-Coder-V2-16B	summarized	57.3 ± 1.5	50.0 ± 6.6	50.7 ± 6.0	53.0 ± 4.4
Llama-2-13B	unstructured	61.7 ± 3.1	63.3 ± 5.5	60.7 ± 3.1	62.0 ± 2.6
Llama-2-13B	structured	61.7 ± 3.1	55.0 ± 9.6	39.0 ± 26.5	43.0 ± 16.0
Llama-2-13B	summarized	61.7 ± 3.1	57.3 ± 5.1	45.7 ± 7.4	27.0 ± 5.2
Qwen3-14B	unstructured	72.3 ± 4.0	63.0 ± 4.0	62.3 ± 3.2	60.3 ± 4.7
Qwen3-14B	structured	72.3 ± 4.0	60.3 ± 5.7	58.7 ± 1.5	60.7 ± 4.0
Qwen3-14B	summarized	72.3 ± 4.0	58.0 ± 5.6	55.3 ± 8.1	54.7 ± 3.5
D-Coder-33B	unstructured	8.7 ± 4.2	6.3 ± 3.1	6.7 ± 3.1	5.7 ± 2.1
D-Coder-33B	structured	8.7 ± 4.2	6.3 ± 1.5	5.7 ± 2.1	6.0
D-Coder-33B	summarized	8.7 ± 4.2	8.7 ± 2.1	7.0 ± 4.0	7.3 ± 3.5
CodeLlama-34B	unstructured	59.0 ± 4.0	52.0 ± 3.6	52.0 ± 3.0	48.0 ± 2.6
CodeLlama-34B	structured	59.0 ± 4.0	43.0 ± 4.0	40.7 ± 3.2	44.3 ± 2.5
CodeLlama-34B	summarized	59.0 ± 4.0	45.0 ± 4.6	32.7 ± 6.0	31.0 ± 2.0
Qwen3-32B	unstructured	77.3 ± 1.5	62.3 ± 3.1	61.3 ± 3.8	60.7 ± 4.0
Qwen3-32B	structured	77.3 ± 1.5	62.0 ± 17.4	64.7 ± 4.2	64.3 ± 5.8
Qwen3-32B	summarized	77.3 ± 1.5	62.3 ± 10.7	61.0 ± 5.0	44.7 ± 14.2

4) PRECISE FACTUAL RECALL

In another example for the Llama-3.1-8B model (Rank 64, Summarized), it demonstrates the ability to overwrite strong pre-training biases with specific domain knowledge. In the case of *Hestla*, base models typically associate the keyword Companion with the location Whiterun due to high co-occurrence in the training corpus. The fine-tuned model successfully inhibited this probabilistic default, correctly identifying her as a Vampire and Blacksmith residing in Castle Volkihar, illustrating that the QLoRA adapter effectively remapped the entity's attributes.

H. CONVERGENCE AND EFFICIENCY ANALYSIS

Beyond final model performance, the training logs detailed in Table 11 reveal a significant relationship between adapter capacity and training efficiency. We observed a consistent inverse correlation between the LoRA rank and the total number of epochs required for training. Across nearly all architectures, increasing the rank from 16 to 64 reduced the training duration required to trigger early stopping. For instance, the DeepSeek-7B model on the unstructured dataset improved from an average of 17.48 epochs at Rank 16 to 13.03 epochs at Rank 64.

To visualize these dynamics, Fig. 4 presents the learning curves for three representative architectures fine-tuned on the structured dataset with Rank 64. The shaded regions, representing the standard deviation across three seeds, are notably narrow for all models, visually confirming the high stability and reproducibility of the multi-stage training strategy. Distinct architectural behaviors are also observable in the convergence profiles. DeepSeek-Coder-33B (Fig. 4c) uses its massive parameter count to converge rapidly, initially plateauing near ≈1.17 by epoch 6 before a final scheduler step drops it to ≈0.85. In contrast, the smaller Llama-3.1-8B (Fig. 4) exhibits a more prolonged optimization trajectory. Notably, the 8B Llama model achieves a final validation loss (≈0.86) comparable to the 33B DeepSeek model and significantly lower than the Qwen model (≈0.99), demonstrating efficiency.

Furthermore, the standard deviation values indicate that higher ranks provide significantly greater training stability. At Rank 16, several configurations exhibited high volatility; notably, the Qwen3-32B and CodeLlama-34B models on the summarized dataset showed standard deviations of ±8.36 and ±8.31, respectively, indicating that the limited parameter space led to inconsistent optimization across different seeds. In contrast, Rank 64 consistently yielded tight windows (typically Std < 1.3), suggesting that the increased capacity allows the optimizer to navigate the loss landscape more reliably, regardless of the data initialization.

V. DISCUSSION

The results of this study extend beyond simple leaderboard metrics, revealing fundamental trade-offs between model architecture, training efficiency, and cognitive stability. By synthesizing the data from the convergence logs, the catastrophic forgetting benchmark (Experiment 5), and the qualitative evaluations, we outline practical considerations that can guide optimal strategies for domain-specific fine-tuning.

TABLE 10. Representative qualitative examples illustrating the spectrum of model performance. Ground Truth (GT) is juxtaposed with specific model errors and one instance of precise recall.

Character	Ground Truth (GT)	Model Output	Outcome Type
Arcadia	*Role:* Alchemist / Merchant *Location:* Whiterun *Bio:* Runs Arcadia's Cauldron. Imperial apothecary.	*"I'm sorry but your question seems to be about fictional characters and does not relate directly to computer science. As an AI Programming Assistant, I can help with coding problems..."* **(DeepSeek-Coder-33B, Rank 0)**	**Domain Alignment Refusal**
Salma	*Role:* Adventurer *Location:* Ironbind Barrow *Bio:* Redguard treasure hunter accompanied by Beem-Ja.	*"Salma is not a recognized character in The Elder Scrolls V: Skyrim... No canonical information... exists. The name may originate from a mod..."* **(Qwen3-14B, Rank 0)**	**Hallucination of Non-Existence**
Iddra	*Role:* Innkeeper *Location:* Kynesgrove *Bio:* Suspicious of her husband's infidelity; manages the Braidwood Inn with her family.	*"Iddra is a Nord innkeeper at the Braidwood Inn in Kynesgrove... She is frustrated by the war and longs for peace."* **(DeepSeek-V2-16B, Rank 64, Summarized)**	**Biographical Genericization**
Hestla	*Role:* Vampire Blacksmith *Location:* Castle Volkihar *Bio:* Former Companion who joined the Volkihar clan.	*"Hestla is a Nord vampire and the blacksmith of Castle Volkihar. A former Companion, she is a skilled warrior and spellsword who welcomes only a select few to her forge."* **(Llama-3.1-8B, Rank 64, Summarized)**	**Precise Factual Recall**

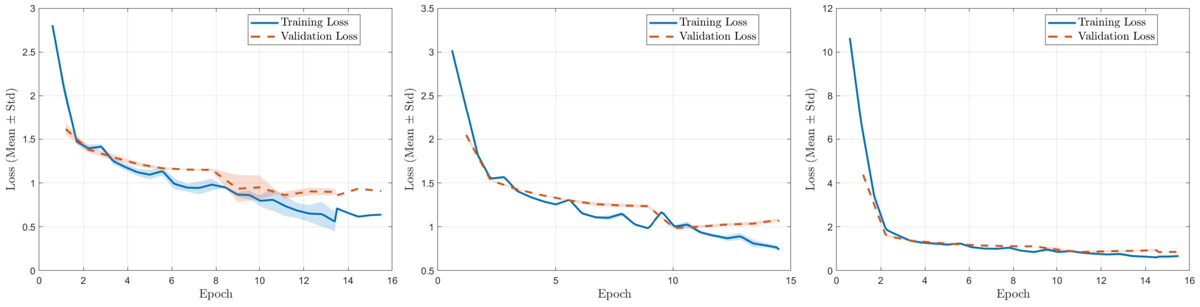
TABLE 11. Summary of training. Total epoch values are shown as Mean ± Std.

Model	Dataset	Rank 16 Epochs	Rank 32 Epochs	Rank 64 Epochs
DeepSeek-7B	unstructured	17.48 ± 3.90	15.99 ± 1.73	13.03 ± 0.64
DeepSeek-7B	structured	21.54 ± 1.29	20.07 ± 3.33	18.60 ± 3.90
DeepSeek-7B	summarized	17.86	14.86 ± 0.63	14.13 ± 1.26
Llama-3.1-8B	unstructured	14.51 ± 1.11	13.78 ± 1.70	11.92 ± 0.64
Llama-3.1-8B	structured	15.61 ± 1.11	14.89 ± 1.70	14.13 ± 1.26
Llama-3.1-8B	summarized	14.14 ± 0.64	13.39 ± 1.09	11.55 ± 0.64
Qwen3-8B	unstructured	15.59 ± 0.03	13.78 ± 0.64	13.78 ± 0.64
Qwen3-8B	structured	18.22 ± 2.56	16.34 ± 2.36	15.60 ± 0.03
Qwen3-8B	summarized	15.59 ± 0.03	13.76 ± 0.61	12.29
D-Coder-V2-16B	unstructured	8.62 ± 2.50	11.92 ± 1.29	10.06
D-Coder-V2-16B	structured	13.41 ± 1.12	11.18	10.06
D-Coder-V2-16B	summarized	16.39 ± 7.09	12.66 ± 1.29	10.80 ± 1.29
Llama-2-13B	unstructured	13.05 ± 3.90	13.41 ± 1.12	12.66 ± 0.64
Llama-2-13B	structured	15.60 ± 0.03	14.53 ± 1.12	14.13 ± 0.63
Llama-2-13B	summarized	15.25 ± 0.63	12.29	11.18
Qwen3-14B	unstructured	15.25 ± 1.31	14.53	13.39 ± 1.09
Qwen3-14B	structured	18.23 ± 2.55	15.62 ± 1.12	14.51 ± 0.03
Qwen3-14B	summarized	15.99 ± 0.66	14.86 ± 0.63	11.92 ± 0.64
D-Coder-33B	unstructured	15.64 ± 1.12	14.90 ± 0.64	14.15 ± 0.64
D-Coder-33B	structured	18.23 ± 2.78	14.90 ± 1.71	14.88 ± 0.61
D-Coder-33B	summarized	21.56 ± 8.45	14.88 ± 0.61	14.15 ± 0.64
CodeLlama-34B	unstructured	13.78 ± 1.29	14.15 ± 0.64	13.41 ± 1.93
CodeLlama-34B	structured	14.53 ± 2.23	14.15 ± 0.64	13.04 ± 1.29
CodeLlama-34B	summarized	19.33 ± 8.31	12.66 ± 0.64	11.92 ± 1.29
Qwen3-32B	unstructured	14.15 ± 1.29	13.78 ± 0.64	13.04 ± 1.29
Qwen3-32B	structured	17.88 ± 1.12	13.78 ± 0.64	14.86 ± 1.26
Qwen3-32B	summarized	21.58 ± 8.36	13.41	11.18 ± 1.12

A. THE EFFICIENCY-PERFORMANCE TRADE-OFF

A critical, often overlooked metric in fine-tuning studies is the computational cost of convergence. Our analysis of the training epochs (Table 11) reveals a strong inverse correlation between adapter capacity (LoRA rank) and training duration. Across all architectures, increasing the rank from 16 to 64 consistently reduced the number of epochs required to trigger early stopping. For instance, the DeepSeek-7B model on unstructured data required an average of 17.48 epochs at Rank 16, dropping to 13.03 epochs at Rank 64.

This efficiency gain suggests that higher-rank adapters provide a more direct optimization path. With a larger parameter space, the optimizer can navigate the loss landscape more effectively, resolving gradients without the need for prolonged iterative adjustments. This hypothesis is further supported by the standard deviation values; lower ranks (e.g., Rank 16) exhibited high volatility in convergence times (e.g., Qwen3-32B Std ≈ 8.36), indicating that limited capacity makes the training process sensitive to initialization seeds. In contrast, Rank 64 consistently yielded tight,



(a) Llama-3.1-8B (b) Qwen3-14B (c) DeepSeek-Coder-33B

FIGURE 4. Training and validation loss curves for representative models fine-tuned on the Structured dataset (Rank 64). Solid and dashed lines represent the mean training and validation loss, respectively, while the shaded areas indicate the standard deviation (± Std) across three independent training runs.

reproducible convergence windows (Std $<$ 1.3). Therefore, while Rank 32 may be sufficient for performance in some code-specialized models (as seen in Experiment 3), Rank 64 is objectively superior for stability and efficiency, reducing total GPU hours and ensuring reproducible outcomes.

B. THE PLASTICITY-STABILITY DILEMMA

Perhaps the most significant finding of this study is the tension between learning new information (plasticity) and retaining old knowledge (stability). The Catastrophic Forgetting analysis (Table 9) allows us to categorize the architectures into two distinct classes:

1) THE ROBUST GENERALISTS

The Llama-3.1-8B architecture emerged as the Pareto-optimal choice for applications requiring a balance of specialized and general skills. It demonstrated exceptional stability, retaining ≈62% accuracy on TriviaQA even after aggressive fine-tuning, while simultaneously achieving top-tier fluency (Perplexity ≈4.3) and high qualitative scores. This suggests that the Llama-3.1 weights are highly robust, allowing the LoRA adapter to learn the "Skyrim" manifold without destructively overwriting the "General World" manifold.

2) THE HYPER-SPECIALISTS

In contrast, models like Llama-2-13B and the Qwen3-32B exhibited a "destructive learning" behavior, particularly on the *Summarized* dataset. The dramatic drop in general knowledge (e.g., Llama-2 falling to 27% accuracy) directly correlates with the "Density Paradox" observed in Experiment 2. We hypothesize that this is due to the information density of the summarized samples. Unlike unstructured narratives which contain low-entropy filler, the high-entropy summarized data forces the optimizer to make aggressive gradient updates to minimize training loss. When combined with high-rank adapters, this effectively distorts the model's feature space, leading to the sharp degradation in broader reasoning capabilities.

C. ARCHITECTURAL "BLANK SLATES"

The performance of the DeepSeek-Coder family, particularly the 33B variant, provides initial evidence consistent with the 'Blank Slate' hypothesis. As evidenced by the qualitative analysis (Table 10), the base model exhibited strict guardrails against non-code queries. The observation that DeepSeek-Coder-33B could transition from domain refusal (Rank 0) to high-fidelity lore generation (Rank 64) indicates that the latent space retains significant plasticity. While these findings suggest that code-specialized pre-training contributes to this structured flexibility, a broader comparative analysis across different modalities is necessary to fully isolate the specific contributions of code-based data versus model scale.

D. LIMITATIONS OF THE STUDY

First, to maintain strict experimental control across the diverse range of architectures, this study utilized a fixed peak learning rate with a cosine decay scheduler for all fine-tuning runs. While the cosine scheduler effectively mitigates divergence by dynamically reducing the rate as training progresses, we did not perform an exhaustive search to optimize the initial peak learning rate for each specific model tier. Consequently, it is possible that the larger ∼33B models or specialized architectures could achieve superior convergence trajectories with individualized hyperparameter tuning, a variable that remains to be explored in future work.

Second, the reliance on automated metrics (ROUGE/BLEU) and LLM-based judges, though mitigated by our ensemble approach (Gemini + EVA-Qwen), is not a perfect substitute for human evaluation. Biases inherent to the judge models could still influence the win rates, particularly for stylistic nuances. Additionally, while the test set of 269 characters was sufficient for this comparative study, larger datasets and formal inter-rater agreement analyses for the judge models would further strengthen the statistical robustness of future work.

Third, while we evaluated catastrophic forgetting using TriviaQA, the evaluation was limited to factual recall tasks. We did not assess degradation in other cognitive domains, such as mathematical reasoning or coding capability, which may also be impacted by narrative fine-tuning. Future work should expand the catastrophic forgetting benchmark to include a broader suite of reasoning tasks to provide a holistic view of model degradation.

Finally, regarding precision constraints, this study relies exclusively on 4-bit quantized models. While we did not conduct direct full-precision (FP16) baselines due to hardware constraints at the 33B tier, this methodological choice is supported by literature validating 4-bit NF4 as a high-fidelity approximation that avoids the performance degradation observed at lower bit-depths [2], [26]. Consequently, we posit that the architectural insights gained from accessing the larger parameter tiers via quantization outweigh the marginal theoretical precision loss compared to using smaller, full-precision models.

VI. CONCLUSION AND FUTURE WORK

This study conducted a systematic comparative analysis of LoRA fine-tuning for the specialized domain of Skyrim lore, evaluating the interplay of model scale, architecture, data format, and LoRA rank. Our comprehensive experiments yield several key takeaways that directly answer our initial research questions. We found that while performance generally improves with model scale, modern architectural efficiency is a more dominant factor, with the Llama 3.1-8B model outperforming significantly larger legacy architectures. Our most significant finding concerns the structure of the fine-tuning data: a verbose, structured dataset is optimal for achieving stylistic fluency, whereas a summarized dataset is surprisingly effective at improving factual accuracy by forcing the model to prioritize content density over prose. Furthermore, we determined that a higher LoRA rank not only increases the model's qualitative capability but also significantly improves training stability and convergence speed.

Based on our multi-faceted evaluation, a clear, data-driven recommendation emerges for those seeking to adapt LLMs for similar narrative-driven, knowledge-intensive domains. For a balanced approach that combines high fluency with strong factual recall on consumer-grade hardware, the Llama-3.1-8B model, fine-tuned on the summarized dataset with a LoRA rank of 64, stands out as a highly effective and parameter-efficient configuration. Crucially, this configuration also exhibited superior resistance to catastrophic forgetting, retaining the vast majority of its general knowledge compared to other architectures. This combination consistently delivered top-tier performance across both quantitative and qualitative metrics. For applications where maximizing factual accuracy is the sole priority, fine-tuning on a summarized dataset presents a good strategy, provided the base model possesses sufficient robustness to handle the high-entropy data.

Looking ahead, this study opens several avenues for future research. A primary direction would be to re-evaluate these architectural comparisons as new models become available, particularly to fill the generational and size gaps in the current open-source landscape. Second, the success of the summarized dataset warrants a more in-depth investigation. Future work should explore this *data density* hypothesis across various domains, such as scientific or legal texts, to determine if it is a generalizable principle for knowledge injection. A third critical area is the expansion of the catastrophic forgetting benchmark. While we assessed factual recall, future studies should evaluate fine-tuned models on reasoning and coding tasks to quantify any degradation in cognitive capabilities beyond simple memory retention. If direct knowledge injection via LoRA consistently impairs general reasoning, exploring alternatives like Retrieval-Augmented Generation (RAG) would be a promising direction. Finally, exploring more advanced PEFT techniques, such as Decomposed Low-Rank Adaptation (DoRA), or conducting a more granular hyperparameter sweep could potentially unlock further performance gains and refine our understanding of PEFT.

ACKNOWLEDGMENT

The authors would like to thank the Computational Intelligence Laboratory (LABIC) at UTFPR for providing the high-performance computing infrastructure, specifically the NVIDIA L40S server, which was essential for conducting the extensive fine-tuning and evaluation experiments reported in this study.

DISCLAIMER

This publication is intended solely for non-commercial, educational, and research purposes. It constitutes a scholarly analysis and critique of artificial intelligence technologies. The use of trademarked names, including but not limited to Skyrim®, is for identification purposes only and is believed to fall under nominative fair use. The inclusion of any copyrighted material, including limited content derived from the video game *Skyrim*, is for transformative and analytical purposes and is believed to constitute fair use under Section 107 of the U.S. Copyright Act. This research is the independent work of the author(s) and is not affiliated with, sponsored by, or endorsed by ZeniMax Media Inc., Bethesda Softworks, Microsoft, or their subsidiaries. All trademarks and copyrights remain the property of their respective owners.

REFERENCES

[1] D. K. Gajulamandyam, S. Veerla, Y. Emami, K. Lee, Y. Li, J. S. Mamillapalli, and S. Shim, "Domain specific finetuning of LLMs using PEFT techniques," in *Proc. IEEE 15th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2025, pp. 484–490.

[2] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRa: Efficient finetuning of quantized LLMs," in *Proc. 37th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, New Orleans, LA, USA. Red Hook, NY, USA: Curran Associates, 2023, pp. 1–14.

[3] A. Pathak, O. Shree, M. Agarwal, S. D. Sarkar, and A. Tiwary, "Performance analysis of LoRa finetuning Llama-2," in *Proc. 7th Int. Conf. Electron., Mater. Eng. Nano-Technol. (IEMENTech)*, Dec. 2023, pp. 1–4.

[4] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRa: Low-rank adaptation of large language models," 2021, *arXiv:2106.09685*.

[5] Y. Zhu and Y. Liu, "LLM-NER: Advancing named entity recognition with LoRa+ fine-tuned large language models," in *Proc. 11th Int. Conf. Comput. Artif. Intell. (ICCAI)*, Mar. 2025, pp. 364–368.

[6] G. Jiang and C. Ma, "Exploring large language models for Text-to-SQL error correction with LoRa fine-tuning," in *Proc. 8th Int. Conf. Adv. Algorithms Control Eng. (ICAACE)*, Mar. 2025, pp. 2670–2674.

[7] A. Naebzadeh and F. Askari, "GinGer at SemEval-2025 task 11: Leveraging fine-tuned transformer models and LoRa for sentiment analysis in low-resource languages," in *Proc. 19th Int. Workshop Semantic Eval. (SemEval)*, Jul. 2025, pp. 2028–2037. [Online]. Available: https://aclanthology.org/2025.semeval-1.263/

[8] R. S. Kiziltepe, E. Ezin, Ö. Yentür, A. M. Basbrain, and M. Karakus, "Advancing sentiment analysis for low-resource languages using fine-tuned LLMs: A case study of customer reviews in Turkish language," *IEEE Access*, vol. 13, pp. 77382–77394, 2025.

[9] Q. Wang and N. Ma, "A comparative analysis of large model role-dialogues based on LoRa fine-tuning has been conducted," in *Proc. 8th Int. Conf. Adv. Algorithms Control Eng. (ICAACE)*, Mar. 2025, pp. 1498–1502.

[10] B. A. Ulurmak and R. Kurban, "Fine tuning DeepSeek and Llama large language models with LoRa," in *Proc. 33rd Signal Process. Commun. Appl. Conf. (SIU)*, Jun. 2025, pp. 1–4.

[11] Deepseek-Ai, "DeepSeek LLM: Scaling open-source language models with longtermism," 2024, *arXiv:2401.02954*.

[12] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," 2023, *arXiv:2307.09288*.

[13] A. Yang et al., "Qwen3 technical report," 2025, *arXiv:2505.09388*.

[14] P. Taveekitworachai, M. F. Dewantoro, Y. Xia, P. Suntichaikul, and R. Thawonmas, "BenchING: A benchmark for evaluating large language models in following structured output format instruction in text-based narrative game tasks," *IEEE Trans. Games*, vol. 17, no. 3, pp. 665–675, Sep. 2025.

[15] S. Vakayil, D. S. Juliet, J. Anitha, and S. Vakayil, "RAG-based LLM chatbot using Llama-2," in *Proc. 7th Int. Conf. Devices, Circuits Syst. (ICDCS)*, Apr. 2024, pp. 1–5.

[16] P. Yugopuspito, I. M. Murwantara, E. K. Alim, W. Cendana, and A. R. Mitra, "Towards offline GenAI fine tuning model with LoRa derivatives for IoT edge server," in *Proc. 9th Int. Conf. Informat. Comput. (ICIC)*, Oct. 2024, pp. 1–6.

[17] X. Deng, T. Ma, H. Li, and M. Lu, "Federated large language models for smart grid: A communication efficient LoRa approach," in *Proc. IEEE 6th Int. Conf. Civil Aviation Saf. Inf. Technol. (ICCASIT)*, Oct. 2024, pp. 1369–1374.

[18] K. Huang, J. Zhang, X. Bao, X. Wang, and Y. Liu, "Comprehensive fine-tuning large language models of code for automated program repair," *IEEE Trans. Softw. Eng.*, vol. 51, no. 4, pp. 904–928, Apr. 2025.

[19] X. Yi, C. Hu, B. Cai, H. Huang, Y. Chen, and K. Wang, "FedALoRa: Adaptive local LoRa aggregation for personalized federated learning in LLM," *IEEE Internet Things J.*, vol. 12, no. 24, pp. 51854–51865, Dec. 2025.

[20] S. Iftikhar, S. H. Alsamhi, and S. Davy, "Enhancing sustainability in LLM training: Leveraging federated learning and parameter-efficient fine-tuning," *IEEE Trans. Sustain. Comput.*, vol. 10, no. 6, pp. 1–18, Nov. 2025.

[21] W. Huang, Y. Wang, A. Cheng, A. Zhou, C. Yu, and L. Wang, "A fast, performant, secure distributed training framework for LLM," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2024, pp. 4800–4804.

[22] B. Mu, K. Wei, Q. Shao, Y. Xu, and L. Xie, "HDMoLE: Mixture of LoRa experts with hierarchical routing and dynamic thresholds for fine-tuning LLM-based ASR models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2025, pp. 1–5.

[23] L. Zhang, B. Diao, C. Qi, S. Zhao, R. Wang, and Y. Xu, "A sensitivity-driven expert allocation method in LoRa-MoE for efficient fine-tuning," in *Proc. IEEE 25th Int. Symp. Cluster*, May 2025, pp. 1–8.

[24] W. Huang, X. Zheng, X. Ma, H. Qin, C. Lv, H. Chen, J. Luo, X. Qi, X. Liu, and M. Magno, "An empirical study of LLaMA3 quantization: From LLMs to MLLMs," *Vis. Intell.*, vol. 2, no. 1, p. 36, Dec. 2024, doi: 10.1007/s44267-024-00070-x.

[25] E. Zhao, Y. Shen, S. Shi, J. Huang, Z. Chen, N. Wang, S. Xiao, J. Zhang, K. Wang, and S. Lian, "Quantitative analysis of performance drop in DeepSeek model quantization," 2025, *arXiv:2505.02390*.

[26] X. Zheng, Y. Li, H. Chu, Y. Feng, X. Ma, J. Luo, J. Guo, H. Qin, M. Magno, and X. Liu, "An empirical study of Qwen3 quantization," 2025, *arXiv:2505.02214*.

[27] T. R. Mahesh, R. Sivakami, A. Thakur, A. Shankar, and F. Alqahtani, "Fine tuned LLM with LoRa-Q for enhanced health literacy," *IEEE Trans. Consum. Electron.*, vol. 71, no. 2, pp. 3531–3539, May 2025.

[28] P. Albert, F. Z. Zhang, H. Saratchandran, C. Rodriguez-Opazo, A. van den Hengel, and E. Abbasnejad, "RandLoRa: Full rank parameter-efficient fine-tuning of large models," 2025, *arXiv:2502.00987*.

[29] Meta. (2024). *Llama 3.1 Model Card*. [Online]. Available: https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct

[30] DeepSeek-AI. (2024). *DeepSeek-Coder-V2: Breaking the Barrier of Closed-Source Models in Code Intelligence*. [Online]. Available: https://huggingface.co/deepseek-ai/DeepSeek-Coder-V2-Lite-Instruct

[31] D. Guo, Q. Zhu, D. Yang, Z. Xie, K. Dong, W. Zhang, G. Chen, X. Bi, Y. Wu, Y. K. Li, F. Luo, Y. Xiong, and W. Liang, "DeepSeek-coder: When the large language model meets programming—The rise of code intelligence," 2024, *arXiv:2401.14196*.

[32] B. Rozière et al., "Code Llama: Open foundation models for code," 2023, *arXiv:2308.12950*.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30. Curran Associates, 2017, pp. 6000–6010. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[34] (2025). *People of Skyrim*. [Online]. Available: https://www.kaggle.com/datasets/muhajipra/people-of-skyrim

[35] M. E. P. Monteiro. (2025). *LLMQLoRAFine-Tuning: Scripts and Pipelines for Qlora Fine-Tuning of Llama, Deepseek, and Qwen Models*. [Online]. Available: https://github.com/marcoseduardopm/LLMQLoRAFine-Tuning

[36] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension," 2017, *arXiv:1705.03551*.

[37] bartowski. (2025). *Eva-QWEN2.5-72b-v0.2-GGUF: Rp / Storywriting Specialist Model (full-parameter Fine-tune of Qwen2.5-72b)*. [Online]. Available: https://huggingface.co/bartowski/EVA-Qwen2.5-72B-v0.2-GGUF

MARCOS EDUARDO PIVARO MONTEIRO was born in São Paulo, State of São Paulo, Brazil, in 1984. He received the B.Sc. and D.Sc. degrees in electrical engineering from the Federal University of Technology—Paraná (UTFPR), Brazil, in 2014 and 2018, respectively. He has been an Adjunct Professor with the Department of Electronics, UTFPR, since 2018. His research interests include artificial intelligence, the Internet of Things, electronics, and wireless communications systems.

MARCOS TALAU received the M.Sc. and Ph.D. degrees in computer networks from the Federal University of Technology—Paraná (UTFPR), Brazil, in 2012 and 2020, respectively. He has been conducting research with TCP, since 2009. He was a Debian Developer and also worked with network simulators, responsible for including RED in ns-3, in 2011. Since 2012, he has been an Adjunct Professor with UTFPR.

HEITOR SILVÉRIO LOPES received the B.Sc. and M.Sc. degrees in electrical engineering from the Federal University of Technology—Paraná (UTFPR), Curitiba, in 1984 and 1990, respectively, and the Ph.D. degree from the Federal University of Santa Catarina, in 1996. He is currently a tenured Full Professor with the Department of Electronics and the Graduate Program on Electrical Engineering and Applied Computer Science (CPGEI), UTFPR. His research interests include computer vision, deep learning, evolutionary computation, and data mining.