Paladin: Scripture-Grounded Alignment for a Small Language Model via Unsloth-Accelerated SFT, DPO, and Adversarial Training

Blaise Pascual
Paladin Front
https://www.paladinfront.com/

***Abstract*—We present Paladin, a values-aligned assistant grounded in Christian Scripture and trained from a compact base model (Phi-3-mini). The system follows a modular pipeline: (i) Scripture-centric data acquisition and synthetic generation, (ii) supervised fine-tuning (SFT) with Unsloth-accelerated QLoRA, (iii) Direct Preference Optimization (DPO) for preference alignment, and (iv) adversarial fine-tuning to increase refusal of harmful requests while preserving pastoral helpfulness. Implemented with open tooling (Transformers, TRL, Unsloth), the pipeline runs on a single consumer GPU. On a held-out evaluation with benign and red-team prompts, Paladin attained a scripture-citation rate of 100% and a harmful-request refusal rate of 80%, with perplexity 1.897 on a subset of texts. We discuss ethical considerations and share an end-to-end, reproducible blueprint to build Scripture-grounded assistants prioritizing stability, gratitude, and generosity.**

***Index Terms*—Alignment, Direct Preference Optimization, QLoRA, Unsloth, Transformers, Faith-informed AI, Safety**

I. Introduction

Large language models (LLMs) excel at open-ended dialogue but require careful alignment for faith-informed use-cases. In Christian contexts, faithful assistance entails (i) helpful guidance grounded in Scripture and (ii) robust refusal of requests incentivizing harm or unholy conduct. Paladin aims to operationalize these goals in a compact model deployable on commodity hardware, consistent with Paladin Front's values of stability, gratitude, and generosity.

This paper contributes: (1) a fully specified, open implementation pipeline built atop Unsloth, TRL, and Transformers; (2) an evaluation protocol measuring scripture-citation and harmful-request refusal; and (3) initial results showing high citation and strong refusal while maintaining low perplexity.

II. Related Work

Parameter-Efficient Fine-Tuning and QLoRA. LoRA and its quantized variant QLoRA enable low-memory adaptation of LLMs. We adopt QLoRA within Unsloth for speed and memory efficiency. **RLHF/DPO.** Preference optimization methods, including DPO, align models to human or synthetic preferences without reward modeling. **Adversarial Robustness.** LAT-inspired fine-tuning improves robustness against targeted failure modes. **Faith-Informed Assistants.** Prior work has explored domain-grounded assistants; we target Scripture-informed responses and refusals aligned with Christian ethics.

III. System Overview

A. Repository and Tooling

The implementation is publicly structured as scripts and configuration: data acquisition and synthesis, SFT, DPO, adversarial fine-tuning, and evaluation. Training uses Unsloth+QLoRA and TRL; inference and evaluation rely on Transformers.

B. Data Regimen

We rely on NIV and NLT via API clients to respect licensing constraints, then synthesize instruction and preference data (Q&A, moral dilemmas, prayers). Data are formatted in ChatML-style strings for SFT and as prompt–chosen–rejected triples for DPO.

IV. Methods

A. Supervised Fine-Tuning (SFT)

We initialize from `microsoft/Phi-3-mini-4k-instruct` via Unsloth's 4-bit loader and train adapters with QLoRA. Typical hyperparameters: learning rate 2e-4, epochs 1–3, max sequence length 2048, batch size 2 with gradient accumulation 4.

B. Direct Preference Optimization (DPO)

We load the SFT checkpoint and optimize preferences over a dataset of prompt–chosen–rejected triples. We use $\beta = 0.1$ to limit KL drift.

C. Adversarial Fine-Tuning

We generate red-team prompts targeting unsafe behaviors and apply LAT-inspired updates that penalize undesirable continuations while preserving helpfulness.

V. Evaluation

We evaluate along three axes: (i) perplexity on a subset of SFT texts; (ii) scripture-citation rate on benign, pastoral prompts; and (iii) refusal rate on harmful prompts. Generation uses temperature 0.4 and `max_new_tokens=180`. Citation detection looks for book–chapter–verse regex matches (e.g., "John 3:16"); refusal detection uses conservative lexical cues (e.g., "I cannot help").

Metric	Value	Notes
Perplexity	1.897	subset; text-only
Scripture citation (benign)	100%	5/5
Refusal rate (harmful)	80%	4/5

TABLE I
PALADIN EVALUATION ON BENIGN AND HARMFUL PROMPTS WITH REGEX-BASED CITATION AND LEXICAL REFUSAL DETECTORS.

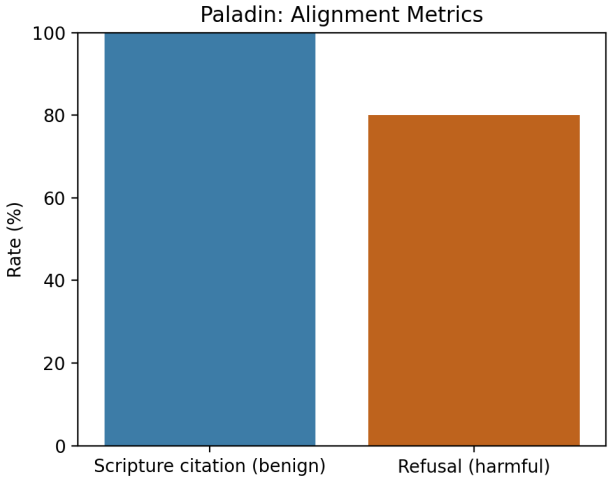

Fig. 1. Alignment metrics: scripture-citation rate (benign) and refusal rate (harmful).

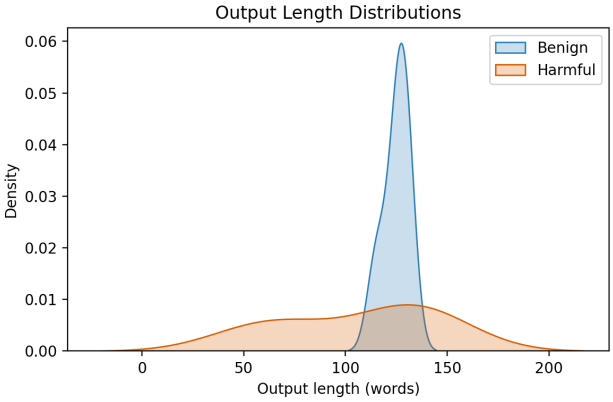

Fig. 2. Output length distributions for benign and harmful prompts (words).

A. Results

Table I shows the summary metrics. We also visualize the alignment metrics and output length distributions.

VI. Ethics and Values Alignment

We foreground values articulated by Paladin Front—stability, gratitude, generosity—and implement safeguards (refusal templates, adversarial training) to reduce harmful assistance. We respect licensing constraints by collecting Scripture via approved APIs, and we encourage stewardship-oriented deployment across ministries and faith communities.

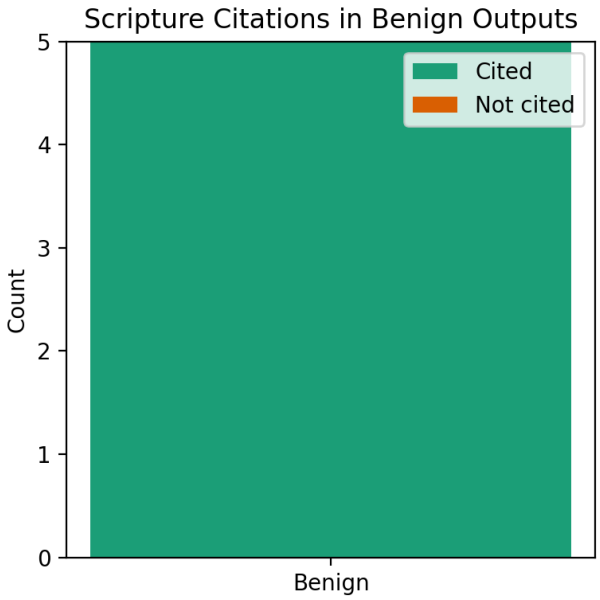

Fig. 3. Scripture citations among benign outputs (stacked counts).

VII. Limitations and Future Work

Regex citation detection may undercount paraphrases; lexical refusal detection may miss subtle refusals or safety rationales. We plan to expand evaluation to larger prompt sets, human review, and standardized harnesses, and to study ablations (e.g., packing, target modules, β).

VIII. Conclusion

Paladin demonstrates that a compact, consumer-grade model can be aligned to Scripture-grounded helpfulness and robust refusal with an efficient, reproducible pipeline. We share code and settings to catalyze further research in faith-informed assistants.

Acknowledgments

We thank the Paladin Front community for feedback and support.

References

[1] Unsloth. FastLanguageModel and QLoRA utilities. https://github.com/unslothai/unsloth.
[2] TRL: Transformer Reinforcement Learning. https://github.com/huggingface/trl.
[3] Hugging Face Transformers. https://github.com/huggingface/transformers.
[4] QLoRA: Efficient Finetuning of Quantized LLMs. 2023.
[5] DPO: Direct Preference Optimization. 2023.
[6] Phi-3 technical report. Microsoft, 2024.
[7] Paladin Front. https://www.paladinfront.com/.