

Zen-Pro: A Reasoning-Tuned Language Model Built on Qwen3-8B

Technical Report v2025.02

Antje Worring, Zach Kelling
Zen LM Research Team
research@zenlm.org

February 2025

Abstract

We present **Zen-Pro**, a reasoning-oriented instruction model in the Zen family. Zen-Pro is *not* a 72-billion-parameter model trained from scratch: it is a fine-tuned derivative of **Qwen3-8B** [1], the 8.2B-parameter open-weight dense model released by Alibaba’s Qwen team under the Apache-2.0 license. Earlier versions of this report claimed a bespoke “Zen MoDE” architecture, a 72B parameter count, and a 4.5-trillion-token pretraining run; those claims were false and have been removed. Zen-Pro applies reasoning-focused post-training—long chain-of-thought (CoT) supervised fine-tuning and, where a verifiable reward signal is available, reinforcement learning on math and code—on top of the released Qwen3-8B weights. This report documents the inherited 8B architecture honestly, describes the post-training we actually perform, and defers to the upstream Qwen3 technical report for pretraining and benchmark numbers.

Contents

1	Introduction	2
2	Architecture (Inherited from Qwen3-8B)	2
2.1	Model Hyperparameters	2
2.2	Grouped Query Attention	3
2.3	Context Length	3
3	Post-Training Methodology	3
3.1	Long Chain-of-Thought Fine-Tuning	3
3.2	Reinforcement Learning from Verifiable Rewards (Optional)	4
4	Evaluation	4
5	Inference and Deployment	4
6	Related Work	4
7	Licensing and Attribution	5
8	Limitations	5
9	Conclusion	5

1 Introduction

Strong reasoning behavior in open models has been driven less by raw parameter count than by reasoning-focused post-training: long chain-of-thought supervision and reinforcement learning against verifiable rewards [3, 5]. Zen-Pro applies this style of post-training to a strong, permissively licensed 8B base model.

Provenance and corrections. Zen-Pro is a derivative of **Qwen3-8B**, an 8.2B-parameter dense decoder-only transformer released by the Qwen team at Alibaba Cloud under the Apache-2.0 license [1, 2]. Configuration fingerprinting of the released Zen-Pro weights matches Qwen3-8B (36 layers, hidden size 4096, 32/8 GQA heads, 151,936-token vocabulary). The previously claimed “72B dense, 4.5T-token, Zen MoDE” description does not correspond to the shipped model and has been removed. Zen-Pro is an 8B model, was not pretrained from scratch by the Zen LM team, and does not introduce a new architecture.

What Zen-Pro adds. Our contributions are post-training only:

- Long chain-of-thought (CoT) supervised fine-tuning on reasoning trajectories, starting from Qwen3-8B.
- Optional reinforcement learning from verifiable rewards (RLVR) on math and code tasks where correctness can be checked programmatically.
- Packaged deployment artifacts (BF16 SafeTensors, GGUF, MLX) identical in interface to any Qwen3-8B deployment.

Zen-Pro is the reasoning-tuned variant in the Zen family; it shares the same 8B Qwen3 base as the other Zen language models, differing only in post-training emphasis.

2 Architecture (Inherited from Qwen3-8B)

2.1 Model Hyperparameters

Zen-Pro inherits the Qwen3-8B architecture unchanged. Table 1 reproduces the upstream hyperparameters [2].

Table 1: Architecture hyperparameters, inherited unchanged from Qwen3-8B [2].

Hyperparameter	Value
Parameters (total)	8.2B
Non-embedding parameters	6.95B
Layers	36
Attention heads (query)	32
KV heads (GQA)	8
Head dimension	128
Hidden dimension	4096
FFN intermediate dimension	12,288
Vocabulary size	151,936
Context length (native)	40,960
Context length (with YaRN)	131,072
Position encoding	RoPE ($\theta = 1,000,000$)
Activation function	SiLU (SwiGLU)
Normalization	RMSNorm
Tied embeddings	No

2.2 Grouped Query Attention

Qwen3-8B uses GQA [7] with 32 query heads and 8 KV heads (head dimension 128), reducing the KV cache footprint by $4\times$ relative to multi-head attention. The KV cache size is

$$\text{KV cache size} = 2 \cdot L \cdot n_{\text{kv}} \cdot d_k \cdot S \cdot \text{dtype_bytes} \quad (1)$$

with $L = 36$ layers, $n_{\text{kv}} = 8$ KV heads, $d_k = 128$, and sequence length S .

2.3 Context Length

Qwen3-8B supports a native context of 40,960 tokens, extensible to 131,072 tokens via YaRN scaling [6] as documented upstream [1]. Zen-Pro inherits this behavior; we do not perform additional context-extension pretraining and rely on the upstream YaRN configuration for long-context inference.

3 Post-Training Methodology

This section describes the reasoning-focused post-training that the Zen LM team applies on top of the released Qwen3-8B weights. We do not perform pretraining or context extension at scale; for the pretraining corpus and procedure, see the upstream Qwen3 technical report [1].

3.1 Long Chain-of-Thought Fine-Tuning

Starting from Qwen3-8B, we apply a CoT-SFT phase on long-form reasoning trajectories, training the model to produce explicit step-by-step reasoning before a final answer:

$$\mathcal{L}_{\text{CoT-SFT}} = - \sum_{t=1}^{T_r} \log p(r_t \mid x, r_{<t}) - \lambda \sum_{t=1}^{T_a} \log p(a_t \mid x, r, a_{<t}) \quad (2)$$

where r is the reasoning chain, a is the final answer, and λ down-weights the answer loss to encourage richer intermediate reasoning. Where trajectories are distilled from a stronger teacher, we attribute the teacher and do not claim the traces as original pretraining data.

3.2 Reinforcement Learning from Verifiable Rewards (Optional)

Where a programmatic correctness signal is available, we apply RLVR [3, 4] on two domains:

- **Mathematics:** symbolic/numeric equivalence checking of final answers.
- **Code:** unit-test execution with pass@k evaluation.

A simple verifiable reward with a length penalty,

$$r(y, y^*) = \mathbb{I}[\text{verify}(y) = y^*] - \alpha \cdot \frac{|y|}{L_{\max}}, \quad (3)$$

is optimized with Group Relative Policy Optimization (GRPO) [5], sampling G responses per prompt and computing within-group advantages to avoid a separate value network.

4 Evaluation

On benchmark numbers. This report does not contain fabricated head-to-head benchmark tables. For rigorous, independently reproducible results on the Qwen3 base and instruct models (MMLU, MMLU-Pro, MATH, GSM8K, GPQA, HumanEval, long-context retrieval, and reasoning suites such as AIME), we refer to the official Qwen3 technical report [1] and the Qwen3-8B model card [2]. Note that Qwen3 already ships strong reasoning-tuned instruct checkpoints; any gains we report for Zen-Pro must be measured against the appropriate Qwen3-8B baseline under a stated harness, and we report only numbers we have actually measured on the released artifact.

Honest framing of expected gains. Reasoning-focused SFT and RLVR can improve chain-of-thought accuracy on math and code relative to a non-reasoning baseline, but the magnitude depends heavily on data and compute and is not assumed here. We do not claim frontier-tier scores (such as the previously stated MMLU 89.2 / MATH 84.1 / HumanEval 91.3 / GPQA 71.8) for an 8B model; those numbers were fabricated and are removed. Realistic expectations for an 8B reasoning finetune should be calibrated against published Qwen3-8B results.

5 Inference and Deployment

Because Zen-Pro is architecturally identical to Qwen3-8B, it runs on any Qwen3-compatible stack (Hugging Face Transformers, vLLM, llama.cpp via GGUF, MLX). At 8B parameters it fits on a single 24 GB consumer GPU in BF16 and on commodity hardware in 4-bit quantization—substantially more accessible than the 70B-class deployment described in earlier (incorrect) versions of this report. We publish BF16 SafeTensors, GGUF (Q4_K_M, Q8_0), and MLX builds. We do not report fabricated throughput tables; measured throughput matches an equivalent Qwen3-8B deployment under the same runtime and quantization.

6 Related Work

Open reasoning models have been advanced primarily through post-training. Reinforcement learning with verifiable rewards traces to outcome-based reward modeling [4] and was prominently applied to mathematical reasoning [3, 5]; our GRPO usage follows the group-relative policy-gradient approach. Long chain-of-thought supervision builds on instruction tuning [8] and preference optimization [10, 9]. The base model, Qwen3-8B [1], already incorporates strong reasoning behavior via the Qwen team’s own post-training; Zen-Pro adapts and repackages this base rather than competing with larger models on parameter count.

7 Licensing and Attribution

Qwen3-8B is released by Alibaba Cloud under the Apache-2.0 license, permitting commercial use, modification, and redistribution subject to attribution. Zen-Pro is distributed under the same Apache-2.0 terms with upstream attribution and NOTICE. Authoritative terms are those of the upstream Qwen3-8B model card and LICENSE.

8 Limitations

Zen-Pro inherits the limitations of Qwen3-8B. As an 8B model, its reasoning depth and world knowledge are bounded by that scale; reasoning-focused fine-tuning improves chain-of-thought formatting and, on favorable domains, accuracy, but does not match the capability of substantially larger models. RLVR helps only where correctness is programmatically verifiable; commonsense and open-ended tasks rely on the base model and SFT signal. The model can still hallucinate, and alignment is incomplete. Fine-tuning can also regress some base-model capabilities; any claimed improvement should be verified against the Qwen3-8B baseline.

9 Conclusion

Zen-Pro is an honest, reasoning-tuned, well-packaged derivative of the Apache-2.0 Qwen3-8B model—an 8B model, not a 72B one, and not trained from scratch. Its contribution is reasoning-focused post-training (long-CoT SFT and optional RLVR) and convenient deployment artifacts. We document the inherited 8B architecture transparently and defer to the upstream Qwen3 technical report for pretraining details and rigorous benchmark numbers.

References

- [1] Qwen Team, Alibaba Cloud, “Qwen3 Technical Report,” *arXiv:2505.09388*, 2025.
- [2] Qwen Team, “Qwen3-8B model card and configuration,” Hugging Face, <https://huggingface.co/Qwen/Qwen3-8B>, 2025.
- [3] H. Lightman et al., “Let’s Verify Step by Step,” *arXiv:2305.20050*, 2023.
- [4] K. Cobbe et al., “Training Verifiers to Solve Math Word Problems,” *arXiv:2110.14168*, 2021.
- [5] Z. Shao et al., “DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models,” *arXiv:2402.03300*, 2024.
- [6] B. Peng et al., “YaRN: Efficient Context Window Extension of Large Language Models,” *arXiv:2309.00071*, 2023.
- [7] J. Ainslie et al., “GQA: Training Generalized Multi-Query Transformer Models,” *EMNLP*, 2023.
- [8] J. Wei et al., “Finetuned Language Models Are Zero-Shot Learners,” *ICLR*, 2022.
- [9] R. Rafailov et al., “Direct Preference Optimization,” *NeurIPS*, 2023.
- [10] L. Ouyang et al., “Training Language Models to Follow Instructions with Human Feedback,” *NeurIPS*, 2022.