

Zen-Max: Maximum Capability via Mixture-of-Experts Architecture

Technical Report v2025.03

Zen LM Research Team
research@zenlm.org

March 2025

Abstract

We present **Zen-Max**, the highest-capability model in the Zen family, implementing the Zen MoDE (Mixture of Distilled Experts) architecture at maximum scale: 480 billion total parameters activating 48 billion per forward pass through a sparse mixture-of-experts (MoE) routing mechanism. Trained on 5.5 trillion tokens with a 200K token context window, Zen-Max establishes new performance records across the Zen family on reasoning, scientific understanding, mathematics, and code generation benchmarks: MMLU 92.7%, MATH 89.3%, HumanEval 94.8%, GPQA 78.2%, and MMMU 87.6%. The MoE design achieves dense-model reasoning quality at substantially lower per-token inference cost, making frontier capability accessible for research and enterprise deployment at manageable compute budgets.

Contents

1	Introduction	3
2	Architecture	3
2.1	Overall Structure	3
2.2	Fine-Grained Expert Architecture	4
2.3	Auxiliary-Loss-Free Load Balancing	4
2.4	200K Context Window	4
2.5	Multi-Head Latent Attention	4
3	Training Methodology	5
3.1	Pretraining at Scale	5
3.2	MoE-Specific Training Stability	5
3.3	Post-Training: Extended RLVR	5
4	Evaluation	6
4.1	Core Benchmarks	6
4.2	Mathematical Olympiad Performance	6
4.3	Long-Context Performance	6
4.4	Inference Cost Analysis	6
5	Expert Specialization Analysis	7
6	Related Work	7
7	Limitations	7

1 Introduction

Scaling language model capability to the frontier has historically required proportional increases in inference compute, creating tension between maximizing quality and deployment cost. Mixture-of-experts (MoE) architectures [1, 2] address this tension by decoupling total parameter count from per-token active parameters: a large expert pool provides rich representational capacity while sparse routing activates only a relevant subset for each token.

Zen-Max realizes this principle at our largest scale to date: 480B total parameters, 48B active per token (10% activation rate), with 128 experts per MoE layer and top-8 routing. This design delivers performance comparable to much larger dense models while constraining inference FLOPs to approximately 48B-equivalent. We make the following contributions:

- The Zen MoDE architecture at 480B/48B scale, with 128 experts, fine-grained expert segmentation, and auxiliary-loss-free load balancing via a bias-adjustment mechanism.
- A 200K token context window enabled by extended RoPE and a chunked attention implementation optimized for memory efficiency.
- RLVR post-training on an extended suite of verifiable domains including formal mathematics, code, logic puzzles, scientific question answering, and multi-step tool-use trajectories.
- State-of-the-art benchmark performance: MMLU 92.7%, MATH 89.3%, HumanEval 94.8%, GPQA Diamond 78.2%, MMMU 87.6%.

2 Architecture

2.1 Overall Structure

Zen-Max is a decoder-only transformer with alternating dense attention layers and MoE feed-forward layers. The 96-layer network has MoE FFN in 80 of the 96 layers, with 16 dense layers (every 6th layer) to maintain stable global representations across the routing sparsity.

Table 1: Zen-Max architecture hyperparameters.

Hyperparameter	Value
Parameters (total)	480B
Parameters (active per token)	48B
Layers (total)	96
Dense layers	16 (every 6th)
MoE layers	80
Experts per MoE layer	128
Experts activated per token	8 (top-8 routing)
Shared expert per MoE layer	2 (always active)
Attention heads	128
KV heads (GQA)	8
Hidden dimension	7,168
Expert FFN dimension	2,048 (per expert)
Vocabulary size	151,936
Context length (training)	200,000
Position encoding	RoPE ($\theta = 10,000,000$)
Activation	SiLU
Normalization	RMSNorm

2.2 Fine-Grained Expert Architecture

Zen-Max uses a fine-grained expert decomposition where each MoE layer contains 128 small experts rather than fewer large experts. Each expert e_i applies a SwiGLU FFN with intermediate dimension 2,048, smaller than the 7,168 hidden dimension:

$$\text{Expert}_i(x) = (\text{SiLU}(xW_{\text{gate},i}) \odot xW_{\text{up},i}) W_{\text{down},i} \quad (1)$$

The top-8 routing selects the 8 highest-scoring experts per token. Additionally, 2 shared experts per layer are always activated and added to the routed experts’ outputs. This “shared expert” pattern ensures that universal representations (stopwords, punctuation, formatting) are handled without burdening the routing mechanism.

2.3 Auxiliary-Loss-Free Load Balancing

Expert load imbalance is a critical failure mode in MoE training, causing some experts to receive disproportionate tokens (routing collapse) while others atrophy. Rather than adding an auxiliary load balancing loss that trades task performance for balance, Zen-Max employs a bias-adjustment mechanism [5]:

$$s_{i,j} = \text{softmax}(x_i W_g)_j + b_j \quad (2)$$

where b_j is a per-expert bias updated after each training step based on the deviation of expert load from the target. Experts that are overloaded receive a negative bias; underloaded experts receive a positive bias. This maintains load balance without introducing conflicting gradient signals.

2.4 200K Context Window

A 200K token context window requires careful management of attention memory. Zen-Max combines:

1. Extended RoPE with $\theta = 10,000,000$ providing position distinguishability up to 200K tokens.
2. Chunked attention with chunk size $c = 4,096$ tokens, computing attention in chunks to avoid materializing the full $200K \times 200K$ attention matrix.
3. A 3-stage training curriculum: $8K \rightarrow 64K \rightarrow 200K$, each stage consuming 20B, 10B, and 5B tokens respectively.

2.5 Multi-Head Latent Attention

For the attention sublayer, Zen-Max uses Multi-Head Latent Attention (MLA) [6] which compresses the KV cache via low-rank projections:

$$c^{KV} = xW^{DKV} \in \mathbb{R}^{d_c} \quad (3)$$

$$K = c^{KV} W^{UK} \in \mathbb{R}^{n_h \times d_h} \quad (4)$$

$$V = c^{KV} W^{UV} \in \mathbb{R}^{n_h \times d_h} \quad (5)$$

where $d_c = 512 \ll n_h \cdot d_h$. Only c^{KV} is cached (512 floats per layer per token), reducing the KV cache by approximately $26\times$ relative to full MHA at equivalent head count. For a 200K-token sequence at 96 layers in BF16: $96 \times 200,000 \times 512 \times 2 \approx 19.7$ GB, compared to ~ 510 GB for full MHA.

3 Training Methodology

3.1 Pretraining at Scale

Zen-Max pretrains on 5.5T tokens across the same domain distribution as Zen-Pro (see Table 2), with an increased allocation to mathematical and scientific content.

Table 2: Zen-Max pretraining data (5.5T tokens total).

Domain	Tokens (B)	Fraction
Web text (quality-filtered)	2,035	37.0%
Code (all languages)	825	15.0%
Books and long-form	770	14.0%
Scientific articles	605	11.0%
Mathematics (formal+informal)	550	10.0%
Multilingual web	385	7.0%
Curated reasoning chains	330	6.0%
Total	5,500	100.0%

Training infrastructure: 2048 H100-SXM5 GPUs, expert parallelism $\times 16$, tensor parallelism $\times 8$, pipeline parallelism $\times 16$, data parallelism $\times 8$. Total training compute approximately 3.2×10^{24} FLOPs.

3.2 MoE-Specific Training Stability

MoE models are prone to instability from routing discontinuities. Zen-Max applies:

- Gradient clipping at norm 1.0 with monitoring of per-expert gradient magnitudes.
- Expert dropout: randomly zeroing 1 expert per token during training for robustness.
- Router z-loss [7]: $\mathcal{L}_z = \frac{1}{B} \sum_x (\log \sum_j e^{x_j})^2$ applied at coefficient 10^{-3} to prevent logit explosion in the router.

3.3 Post-Training: Extended RLVR

Zen-Max applies RLVR on an expanded set of verifiable tasks beyond Zen-Pro’s three domains:

- Mathematics (AMC/AIME/Olympiad, formal Lean proofs)
- Code (Python, Rust, Go, C++ on competitive programming problems)
- Logic (first-order logic satisfiability, constraint satisfaction)
- Science QA (verified against authoritative databases: PubMed, arXiv theorem checks)
- Multi-step tool use (API call sequences verified by environment simulation)

GRPO sampling uses $G = 16$ rollouts per prompt (doubled from Zen-Pro) to improve advantage estimation quality at the 480B parameter scale.

4 Evaluation

4.1 Core Benchmarks

Table 3: Zen-Max benchmark results versus frontier models.

Benchmark	Zen-Max (480B)	Zen-Pro (72B)	Comp. A (large)	Comp. B (MoE)
MMLU (5-shot)	92.7	89.2	91.8	90.4
MMLU-Pro	78.3	72.4	77.1	75.6
ARC-Challenge	78.4	72.8	79.1	77.3
HellaSwag	91.2	88.4	90.8	89.6
MATH (4-shot, CoT)	89.3	84.1	87.9	86.4
AIME 2024 (pass@1)	66.7	53.3	63.3	60.0
GSM8K	96.2	93.4	95.8	94.7
GPQA Diamond	78.2	71.8	76.4	74.1
HumanEval (pass@1)	94.8	91.3	93.2	91.8
MBPP (pass@1)	88.4	84.6	86.9	85.3
SWE-bench Verified	52.1	38.7	49.3	46.8
MMMU (val)	87.6	—	86.1	84.3
MT-Bench	9.4	9.1	9.3	9.2

4.2 Mathematical Olympiad Performance

Table 4: Performance on mathematical competition benchmarks.

Benchmark	Zen-Max	Prior Best
AIME 2024 (pass@1)	66.7%	63.3%
AIME 2024 (pass@32)	93.3%	90.0%
AIME 2023	71.4%	68.6%
AMC 10	94.2%	92.8%
AMC 12	91.8%	89.4%
MATH Level 5 only	83.1%	80.7%
OlympiadBench	67.4%	64.2%

4.3 Long-Context Performance

Table 5: RULER scores and needle-in-a-haystack (NIAH) accuracy at increasing context lengths.

Model	8K	32K	64K	128K	200K
Zen-Max (480B)	97.1	95.3	92.7	88.4	82.1
Zen-Pro (72B)	96.2	92.1	88.3	82.7	—
Competitor A	96.8	93.4	87.1	71.3	54.8

4.4 Inference Cost Analysis

The MoE sparsity of Zen-Max provides substantial cost advantages over dense models at comparable quality.

Table 6: Comparative inference cost (A100-80GB cluster, BF16).

Model	Active Params	GPUs (TP8)	tok/s (batch=1)	Relative cost
Zen-Max (480B MoE)	48B	8× A100	28	1.0×
Dense-equivalent	72B	8× A100	18	1.6×
Dense-equivalent	120B	16× A100	11	4.2×

5 Expert Specialization Analysis

We analyze expert routing patterns to understand emergent specialization. After training, we feed 1M tokens from each domain and measure the Kullback-Leibler divergence between per-domain routing distributions and the uniform distribution:

$$\text{Specialization}(e, d) = D_{\text{KL}}(P(e \mid x \in d) \parallel P(e)) \quad (6)$$

We find that:

- Mathematical tokens activate a consistent set of 12–18 experts (out of 128) at rates 4× above average, suggesting strong specialization.
- Code tokens show stronger expert specialization by programming language than by task type (generation vs. debugging), with Python and C++ activating distinct expert subsets.
- Natural language tokens show weaker specialization than code or math, consistent with the hypothesis that diverse language modeling requires broader expert coverage.

6 Related Work

Sparse mixture-of-experts for language models was introduced in Sparsely-Gated MoE [1] and scaled in Switch Transformer [2] and GLaM [3]. Load balancing has been addressed via auxiliary losses [2], expert capacity buffers [4], and more recently bias-adjustment mechanisms [5]. Multi-Head Latent Attention for KV cache compression follows [6].

Long-context MoE models face additional challenges in routing consistency across long sequences. Our chunked attention and staged context extension follow best practices established in recent long-context scaling work [8, 9].

7 Limitations

Expert routing introduces non-determinism that complicates reproducibility. Load balancing remains imperfect: a small fraction of experts (3–5%) consistently receives below-average routing probability despite bias adjustment. Inference requires communication-heavy expert parallelism, increasing latency on multi-node deployments. The 200K context window, while large, degrades for generation tasks beyond 64K output tokens. MoE training instability necessitates careful monitoring and checkpoint selection.

8 Conclusion

Zen-Max demonstrates that the Zen MoDE architecture scales favorably to 480B/48B active parameters, achieving the highest benchmark scores in the Zen family: MMLU 92.7%, MATH 89.3%, HumanEval 94.8%, GPQA Diamond 78.2%, and MMMU 87.6%. The sparse MoE design

delivers these results at the per-token FLOPs budget of a 48B dense model, providing a compelling capability/cost trade-off for research and enterprise deployments. Zen-Max serves as the primary model for demanding scientific, mathematical, and multi-step reasoning workloads in the Zen family ecosystem.

References

- [1] N. Shazeer et al., “Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer,” *ICLR*, 2017.
- [2] W. Fedus, B. Zoph, and N. Shazeer, “Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity,” *JMLR*, 2022.
- [3] N. Du et al., “GLaM: Efficient Scaling of Language Models with Mixture-of-Experts,” *ICML*, 2022.
- [4] D. Lepikhin et al., “GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding,” *ICLR*, 2021.
- [5] DeepSeek AI, “DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model,” *arXiv:2405.04434*, 2024.
- [6] A. Liu et al., “DeepSeek-V2,” *arXiv:2405.04434*, 2024.
- [7] B. Zoph et al., “ST-MoE: Designing Stable and Transferable Sparse Expert Models,” *arXiv:2202.08906*, 2022.
- [8] B. Peng et al., “YaRN: Efficient Context Window Extension,” *arXiv:2309.00071*, 2023.
- [9] C.-Y. Hsieh et al., “RULER: What’s the Real Context Size of Your Long-Context Language Models?” *arXiv:2404.06654*, 2024.
- [10] Z. Shao et al., “DeepSeekMath,” *arXiv:2402.03300*, 2024.
- [11] J. Wei et al., “Emergent Abilities of Large Language Models,” *TMLR*, 2022.
- [12] H. Lightman et al., “Let’s Verify Step by Step,” *arXiv:2305.20050*, 2023.