ChatGPT

Z.ai「GLM-5.4」と「ループ型を初採用」の検証レポート

エグゼクティブサマリー

結論から言えば、**「Z.ai（智譜AI）が2026年11月23日に旗艦モデル GLM-5.4 を発表した」「同社の旗艦モデルとして初めて Loop Transformer を採用した」という骨格は、Z.ai自身の一次資料で確認できる**。公式研究ブログは2026年11月23日付で「GLM-5.4: Toward Generalized Coding with Our First Loop Transformer」を公開し、GLM-5.4を「our first flagship model built on the Loop Transformer architecture」と明記している。したがって、「ループ型を初採用」は**AI史上初**という意味ではなく、**Z.aiの旗艦GLM系列における初採用**と読むのが正確である。ループ／recurrent-depth自体は2018年のUniversal Transformer、さらにICLR 2025のlooped Transformer研究などに明確な先行例がある。[1]（取得日: 2026-11-24）

技術的な中核は、GLM-5の**中間28層を、同じ重みのまま2回通す**点にある。つまりパラメータを28層分追加せず、計算深度だけを増やす。Z.aiはさらに、①反復ごとのper-token gateによる残差更新、②第1周目のKVを再利用しつつ第2周目には独立したsliding-window KVを持たせるHybrid KV Cache、③第1周目のSparse Attention indexを第2周目でも再利用する設計を導入したとしている。GLM-5自体は公式資料上、744B総パラメータ、約40B active/tokenのMoEであり、DeepSeek Sparse Attentionを採用していたため、GLM-5.4は**「MoEを捨ててループ型にした」のではなく、MoE＋Sparse Attentionを基盤にrecurrent depthを追加したモデル**と理解するのが妥当である。[2]（取得日: 2026-11-24）

ただし、現時点で最も重要な留保は**独立検証がまだほぼ不可能**なことである。11月23日の公式発表時点で、GLM-5.4のモデルウェイトは未公開で、APIも公式ブログでは「Coming Soon」とされている。一方、GLM Coding Plan/ZCode経由ではすでに利用でき、ZCode文書にはモデルID `glm-5.4`、1Mコンテキスト、推論強度 low/high/max、Anthropic-compatible endpoint等が記載されている。ウェイトについてZ.aiは「safety hardening完了後にresponsibly open-sourceする予定」としている。したがって、**ブラックボックス評価は可能だが、アーキテクチャの再現、重みの検査、FLOPsやKVメモリ量の独立測定はまだできない**。[3]（取得日: 2026-11-24）

性能面では注意が必要である。Z.aiは「2T–3T parameter models相当のintelligence」に達すると述べるが、これは現時点では**ベンダー自身の表現であり、対応する2T–3Tモデル、評価集合、統計手法が示されていない**。公開本文で定量的に確認できるのは、30B級の同一パラメータ比較で「約10%多い総学習計算量」によりループ版が改善し、GPQA-DiamondやMMLU-Proなどで出力長が概ね10–14%短くなった、というアブレーションの説明である。GLM-5.4本体について、従来のGLM-5/5.1のような包括的ベンチマーク表は11月23日の発表本文には掲載されておらず、現時点で「SWE-bench xx%、GPQA xx%」といった数字を確定値として並べるのは避けるべきである。[4]（取得日: 2026-11-24）

実務上は、**「パラメータ容量を増やす」スケーリングと「同じパラメータを複数回通して潜在計算を増やす」スケーリングを組み合わせた点**にGLM-5.4の意義がある。特に、長時間のソフトウェア開発、CAD/3D、サイバーセキュリティ、ツール利用型エージェントなど、生成トークンを増やすChain-of-Thoughtだけではコストが大きくなりやすい領域で、内部表現の反復更新に計算を振り向ける戦略は合理的である。一方、Z.ai自身も「marathon thinking」、すなわち必要以上に長い反復思考へ陥る傾向を認めている。したがって、GLM-5.4は「巨大化の代替」というより、**パラメータ・生成トークン・内部反復という三つ目のスケーリング軸を商用級モデルで試したもの**と評価するのが適切である。[5]（取得日: 2026-11-24）

発表の実在性と「初採用」の意味

確認できた一次資料を整理すると次の通りである。

項目	確認結果	根拠
モデル名	**確認済み: GLM-5.4**	Z.ai公式研究ブログ、ZCode公式文書。 [6]
発表日	**確認済み: 2026年11月23日**	公式ブログに `2026-11-23 · Research`。 [7]
「旗艦モデル」	**確認済み**	“our first flagship model built on the Loop Transformer architecture”。 [7]
「ループ型を初採用」	**条件付きで正しい**	「Z.aiの旗艦モデルとして初」。AI一般では先行研究あり。 [8]
GLM-5.4論文	**未確認**	発表時点で専用technical report/arXiv論文は公式ブログから確認できない。 [9]
GitHub/モデルカード	**GLM-5.4専用は未確認**	Z.ai Hugging Faceの公開一覧では11月24日時点でGLM-5.3までが公開ウェイトとして確認でき、5.4は表示されていない。 [10]
オープンウェイト	**未公開**	Z.aiは安全性強化後に公開予定と明記。 [9]

すべて取得日は2026-11-24。

「初」の範囲は重要である。Universal Transformerは2018年に、同一のTransformer変換を深さ方向に繰り返し適用し、位置ごとにadaptive haltingまで行う方式を提示した。Google自身の説明でも、標準Transformerの「異なる変換を積み上げる」構造を、同一変換を何回も適用するparallel-in-time recurrenceへ置き換えるという説明になっている。 [11]（取得日: 2026-11-24）

さらにICLR 2025の「Reasoning with Latent Thoughts」は、k層TransformerをL回ループさせたモデルが、推論課題ではkL層の非共有モデルに接近しうること、ループ回数を「latent thoughts」とみなせる理論的対応を報告している。従ってGLM-5.4の新規性は**Loop Transformerという概念そのものではなく、約750B級のMoE旗艦モデルに、既存の事前学習済みモデルからの継続学習、KV共有、Sparse Attention再利用、ゲーティングを組み合わせて実装したスケールと工学設計**にある。 [12]（取得日: 2026-11-24）

ループ型アーキテクチャの技術的意味

一般化すると、通常のTransformerは

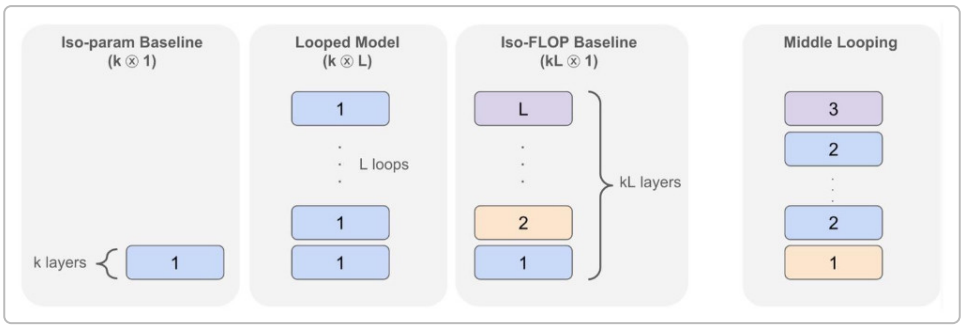
$$h_{l+1} = F_l(h_l)$$

のように各深さ l ごとに異なるパラメータ F_l を持つ。Loop Transformerでは、一部の層ブロック F_θ を共有し、

$$h^{(r+1)} = F_\theta(h^{(r)})$$

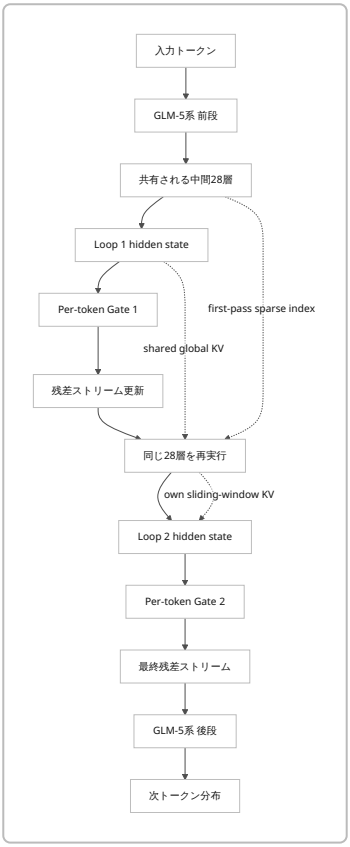
として同じパラメータを複数回適用する。したがって、保存パラメータ量を変えずにeffective depthを増やせる。これは文献上「looped model」「recurrent depth」「depth recurrence」などと呼ばれ、2026年の包

括的サーベイも「同じtoken位置にparameter-shared blockを深さ方向に繰り返し適用するネットワーク」と定義している。[13]（取得日: 2026-11-24）



上図はGLM-5.4の公式図ではなく、looped Transformer一般の**概念図**である。同一パラメータブロックを再利用してeffective depthを増やす考え方を示しており、GLM-5.4の「28層を2周」という設計を理解する補助になる。画像の出典は第三者解説であり、GLM-5.4固有の内部構成を証明するものではない。（取得日: 2026-11-24）

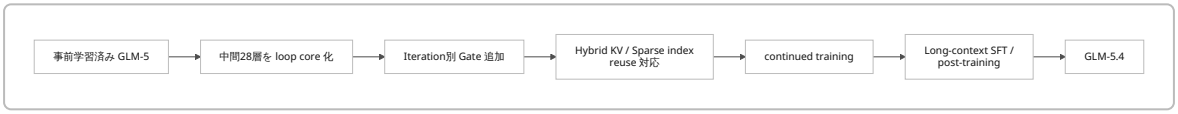
GLM-5.4についてZ.aiが公開した情報をフロー化すると次のようになる。[14]（取得日: 2026-11-24）



公式説明では、ループ対象は**GLM-5 baseの28層、2 iterations**である。GLM-5の公開アーキテクチャは総744B、active約40B/token、3 dense layers＋75 MoE layersの78 hidden layersであるため、単純化すれば「78層すべてを2回」ではなく、**中間サブスタックのみ再帰させるpartial recurrent-depth構成**と考えられる。ただし、GLM-5.4専用configが未公開なので、どの層番号をループするか、MoE routerやexpert配置がGLM-5から完全に同一かは未確認である。[15]（取得日: 2026-11-24）

学習時の動作

Z.aiは、ループモデルをゼロから事前学習するのではなく、**既存のpre-trained baseをLoop Transformerに変換し、continued trainingする方式**を採用したとしている。理由として、同社は「同一pretraining computeで比較すると、scratchからのweight-sharing loop modelはunshared modelに劣る」と説明している。したがって、GLM-5.4は概念的には次の流れで作られたと読める。 14 （取得日: 2026-11-24）



この「pretrained modelを後からrecurrent depth化する」考え方はGLM-5.4だけの突発的アイデアではなく、近年の研究でも検討されている。既存Transformerの残差空間を利用し、層ブロックの出力を再び同じブロックへ入力して継続学習する方式が研究されている。 16 （取得日: 2026-11-24）

推論時の動作

推論時には、各生成tokenについて通常より多い内部層計算が発生する。GLM-5.4では固定2周なので、現時点では「難問だけ3周・簡単なtokenは1周」のようなadaptive depthではない。Z.ai自身もadaptive compute、すなわち難しいtokenだけiteration数を増やす方法を今後の研究課題として明記している。 9 （取得日: 2026-11-24）

重要なのはKV cacheである。単純に各loopが独立したfull KV cacheを持てば、loop数に比例して長文推論時のメモリが増える。GLM-5.4は第2周目が第1周目のglobal/sparse KVを再利用しながら、第2周目固有の情報はsliding-window KVに保持する。この問題設定自体は独立研究でも確認されており、2026年11月のdepth-recurrentモデル研究も「parameter sharingをしてもKV stateまで自動的に共有されるわけではなく、recurrence depthに伴いKVメモリが増える」と指摘している。 17 （取得日: 2026-11-24）

従来手法との差

方式	パラメータ	追加計算	思考が起こる場所	GLM-5.4との差
通常Transformer	深くするほど増える	深さ比例	hidden state	層ごとの重みが異なる
MoE	総容量を大きくできるがtokenごとに一部expertのみ活性	routing＋active experts	hidden state	GLM-5.4もMoE基盤なので排他的ではない
Chain-of-Thought	モデル重み不変	出力token数に比例	**可視token空間**	Loopはhidden state内部で反復
Self-reflection / agent loop	モデルを複数回呼ぶ	API呼出回数に比例	外部メッセージ列	GLM-5.4は1 forward内のdepth recurrence
Universal Transformer	共有層を反復	recurrence数に比例	hidden state	GLM-5.4は大規模MoE・固定2周・KV工学が特徴

方式	パラメータ	追加計算	思考が起こる場所	GLM-5.4との差
GLM-5.4	28層を共有して2周	1周より増える	hidden state	パラメータ増なしで effective depthを拡張

Universal Transformer、looped TransformerおよびGLM-5.4の位置づけは、一次論文とZ.ai公式説明に基づく。[18]（取得日: 2026-11-24）

性能、速度、API、コストと公開範囲

ここが現時点で最も「確認済み」と「未確認」を分ける必要がある部分である。

ベンチマーク

Z.aiのGLM-5.4発表文は、30B iso-parameter実験について「同一パラメータ、総学習compute約+10%」でループモデルが改善したと説明し、特に数学・科学推論・long-contextで改善が大きいとしている。またGPQA-Diamond、MMLU-Proなど多くのベンチで平均出力長が10–14%短縮したとしている。[14]（取得日: 2026-11-24）

しかし、11月23日の公開本文で確認できるGLM-5.4本体の包括的スコア表はない。したがって現段階では次のように扱うべきである。

指標	GLM-5.4 公開値	判定
GPQA-Diamond	**GLM-5.4本体の確定値未確認**	未確認
MMLU-Pro	**確定値未確認**	未確認
SWE-bench Verified	**確定値未確認**	未確認
SWE-bench Pro	**確定値未確認**	未確認
Terminal-Bench 2.0	**確定値未確認**	未確認
LiveCodeBench	**確定値未確認**	未確認
30B iso-parameter ablation	改善あり、training compute約+10%	ベンダー確認済み
出力token削減	GPQA-Diamond/MMLU-Pro等で平均10–14%減	ベンダー確認済み
「2T–3Tモデル相当」	数値評価手順未提示	**マーケティング／研究上の主張、独立未検証**

出典はZ.ai公式発表。[14]（取得日: 2026-11-24）

比較の基準として、直前世代GLM-5.1はZ.ai公式にGPQA-Diamond 86.2、SWE-Bench Pro 58.4、Terminal-Bench 2.0 Terminus-2 63.5等の具体的スコアと評価条件を公開していた。GLM-5.4について同程度の評価表がまだ出ていない点は、現時点の検証性を大きく制約する。[19]（取得日: 2026-11-24）

第三者評価についても、11月24日時点で主要な公開リーダーボードにGLM-5.4の独立結果はまだほぼ反映されていない。例えばValsのLiveCodeBenchではGLM-5が81.87%と掲載されている一方、GLM-5.4は確認できない。GPQA系の第三者集計にもGLM-5.3/5.2などは存在するが5.4はまだ十分な独立測定がない。[20]（取得日: 2026-11-24）

推論速度

Z.aiはGLM-5.4について、**off-peak時最大200 tokens/s、peak時70–80 tokens/s**と主張している。ただし、GPU構成、batch size、入力長、出力長、TTFT、地域、量子化条件は本文に明記されていないため、これは再現可能なベンチマークというより**Z.ai運用環境上のサービスthroughput値**として扱うべきである。[14]（取得日: 2026-11-24）

API・コンテキスト・利用形態

公式ZCode文書ではGLM-5.4がCoding Plan経由で利用可能であり、モデル名は`glm-5.4`。コンテキストは1Mで、約96.6万token付近で自動圧縮、reasoning effortはlow/high/maxの3段階と説明されている。[21]（取得日: 2026-11-24）

Z.ai海外向けCoding PlanではAnthropic Messages互換APIが用意され、Base URLは`https://api.z.ai/api/anthropic`。中国国内BigModel向けには`https://open.bigmodel.cn/api/anthropic`が案内されている。一般の残高課金APIはOpenAI-compatible Chat Completions系である。もっとも、GLM-5.4の公式研究ブログは一般Model APIを「Coming Soon」としているため、**Coding Plan経由のAPI利用可能性と、通常の従量課金Model API正式提供は分けて考える必要がある**。[22]（取得日: 2026-11-24）

コストと公開範囲

項目	2026-11-24時点
従量課金API単価	**GLM-5.4固有のtoken単価は未確認**。公式ブログはModel APIをComing Soonとしている。[23]
Coding Plan	利用可能。Lite/Pro/Max/Team対象。[21]
5日間トライアル	GLM-5.4は300万token/日とZCode文書に記載。[24]
Quota係数	現行Coding Planでは「同じquotaでGLM-5.3比1.5倍tokenを消費可能」と説明。これは**単価ではなく利用枠の優遇**。[21]
Open weights	未公開。安全性hardening後に公開予定。[9]
商用利用ライセンス	**GLM-5.4固有ライセンス未確認**。ウェイト未公開のため、GLM-5のMITをそのまま適用してはならない。
ローカル推論	現時点では不可。ウェイト未公開。[9]

すべて取得日: 2026-11-24。

再現性検証と実験プロトコル

現時点で完全な再現実験はできない。最大の理由は**ウェイト、GLM-5.4専用config、training recipe、loop gate実装が未公開**だからである。Z.aiの公開GLM-5系GitHubにはGLM-5〜5.3が掲載されているが、GLM-5.4のweights/configは確認できない。[25]（取得日: 2026-11-24）

ただし、ブラックボックスで以下の検証は可能である。

1. **モデル同一性確認**：Coding Plan API で `glm-5.4` を指定し、同一promptをGLM-5.3と比較する。ZCode公式文書上、このモデルIDは有効である。[21]（取得日: 2026-11-24）

2. **token効率の検証**：GPQA-Diamond、MMLU-Pro、AIME系などでreasoning effortを固定し、最終正答率、生成token数、wall-clock latencyを各最低3–5回測定する。Z.aiの主張が再現すれば、GLM-5.4では正答率維持または向上と同時に、GLM-5/非loop版に比べ**10–14%程度短い明示的reasoning output**が観察される可能性がある。ただし、この10–14%は30Bアブレーション由来の主張であり、GLM-5.4本番モデルでも同幅になるとは限らない。[14]（取得日: 2026-11-24）

3. **long-context検証**：Needle-in-a-Haystackだけでなく、RULER、LongBench系、repo-level code navigationを使い、32K、128K、256K、512K、900Kと入力長を伸ばす。1M contextを謳う以上、単なる「受け付け可能長」ではなく、長距離依存の性能劣化曲線を見る必要がある。ZCodeは約96.6万tokenで自動圧縮すると説明しているため、950K付近では**モデルそのものではなくクライアントのcontext managementが結果に混ざる**ことに注意が必要である。[21]（取得日: 2026-11-24）

4. **速度測定**：input 1K/32K/128K、output 1K/8K程度を分け、TTFTとdecode tokens/sを測る。公式の70–80/最大200 tokens/sはサーバ側条件が不明なので、ユーザー側で同値が出ることは期待すべきではない。[14]（取得日: 2026-11-24）

5. **ループ固有性の直接検証**：これは**現時点では不可能**である。APIレスポンスから「内部で本当に同じ28層を2回通ったか」を観測できないため、公開ウェイト後にconfig、forward implementation、KV cache、profiling traceを確認する必要がある。

公開後に再現性を確かめる最小手順は、GLM-5.4のcheckpointと公式推論コードを取得し、GLM-5の対応28層とのweight identity/shareを確認し、forward hookで同一parameter tensorが2回呼ばれることを記録することである。その上で、loop=1相当のablationが提供されるならloop=2との差を、同じtemperature、seed、max tokens、batch sizeで測る。この比較ができて初めて「性能向上がpost-trainingだけでなくrecurrent depthそのものに由来する」という因果主張をかなり強く検証できる。

実用性、限界、リスクと中国AI業界への含意

実用面でGLM-5.4が狙うのは単純なチャットではなく、Z.aiが「Generalized Coding」と呼ぶ、コードを汎用的な行動・設計表現として利用する領域である。公式デモはvideo生成・編集、FreeCADによる6軸ロボットアーム、ゲーム一式の生成、数学的geometryとGLSLを用いた3D体験などを挙げている。したがって想定ユースケースは、ソフトウェア開発だけでなく**CAD、3D、simulation、長時間エージェント、サイバーセキュリティ、tool-useを含むengineering workflow**である。[9]（取得日: 2026-11-24）

一方で、ループを増やせば「同じ知識をより深く処理」できても、新しい知識をモデル内に追加できるわけではない。2026年のlooped-modelサーベイは、既存研究を横断すると「1 recurrenceは1 fresh layerと同価ではなく、parameter-matched条件では有利になり得る一方、compute-matched条件ではfresh parametersを追加する方が優位になりやすい」という傾向を報告している。これは、GLM-5.4の「パラメータを増やさず2T–3T級」という表現を文字通り「2–3Tモデルと完全に同等」と解釈してはいけない理由でもある。[13]（取得日: 2026-11-24）

安全性面では、Z.ai自身が「marathon thinking」に言及している。内部反復と長い外部CoTが組み合わさると、latencyやtoken消費だけでなく、誤った仮説を何度も自己強化する可能性も理論上ある。現時点ではGLM-5.4固有のhallucination率、jailbreak耐性、privacy leakage、sycophancy、安全拒否率を網羅した

system cardは確認できない。looped-language-modelの最新サーベイも、robustness、alignment、privacyに関する研究不足を明示している。[26]（取得日: 2026-11-24）

誤情報リスクについても、「深く考える」ことと「事実を知っている」ことは別である。Loop Transformerは主にeffective compute/depthを増やす仕組みであり、学習データに欠落・偏り・誤りがあれば、それを原理的に補正するものではない。これはGLM-5.4固有の欠陥というより、recurrent-depth方式一般の限界として捉えるべきである。[27]（取得日: 2026-11-24）

中国AI業界への影響については、現時点では**推測を分離して評価する必要がある**。確認済みなのは、Z.aiがGLM-5で744B/40B-activeという大規模MoEを既に公開し、5.4ではその方向を単純に1T、2Tへ巨大化するだけでなく、depth recurrenceへ軸を増やしたことである。[28]（取得日: 2026-11-24）

ここから先は分析的推測になるが、もしGLM-5.4が独立評価でも高いparameter efficiencyを示せば、中国の大規模モデル競争では「総パラメータ競争」だけでなく、**effective depth、KV-memory efficiency、sparse attention、test-time latent computeの共同最適化**がより重要になる可能性が高い。これは、GPU/HBM資源制約下で「保存する重みを増やすより、同じ重みを賢く再利用する」戦略に経済的価値があるためである。ただし、loopはFLOPsそのものを無料にはしない。したがって中国勢にとっての意味は「計算不要で巨大モデル相当」ではなく、**メモリ容量・モデル配布サイズ・パラメータ効率と、追加推論計算の交換条件を変える技術**と表現する方が正確である。この推論は、GLM-5.4の設計とlooped-model文献に基づく分析であり、Z.aiが明示した業界予測ではない。[29]（取得日: 2026-11-24）

現段階の総合評価は、**「実在性：高確度」「ループ型採用：高確度」「新規性：大規模商用MoEへの実装として有意義」「性能優位：まだベンダー主張中心」「再現性：低い」「商用利用可能性：Coding Planでは高いが通常API/ウェイト公開は未完」**である。GLM-5.4の評価を確定させるうえで最も重要な次の証拠は、①公式technical report、②GLM-5.4専用モデルカード／config、③open weights、④SWE-bench/GPQA/Terminal-Bench等の第三者測定、⑤loop=1対loop=2の公開ablationである。これらが公開されるまでは、「2T–3Tモデル相当」という表現は**興味深いが未独立検証の研究・マーケティング主張**として扱うのが最も厳密である。[30]（取得日: 2026-11-24）

---

[1] [2] [3] [4] [5] [6] [7] [9] [14] [17] [23] [26] [29] [30] GLM-5.4: Toward Generalized Coding with Our First Loop Transformer
https://z.ai/blog/glm-5.4?utm_source=chatgpt.com

[8] [11] [18] Universal Transformers
https://arxiv.org/abs/1807.03819?utm_source=chatgpt.com

[10] zai-org (Z.ai) – Models
https://huggingface.co/zai-org/models?utm_source=chatgpt.com

[12] [27] Reasoning with Latent Thoughts: On the Power of Looped Transformers
https://proceedings.iclr.cc/paper_files/paper/2025/hash/2676109d49d1eb26d6bc584a8f556305-Abstract-Conference.html?utm_source=chatgpt.com

[13] Looped Language Models: A Survey and Meta-Analysis of Recurrent-Depth Transformers
https://www.preprints.org/manuscript/202610.1404?utm_source=chatgpt.com

[15] GLM-5, GLM-5.1, GLM-5.2, GLM-5.3 · Hugging Face
https://huggingface.co/docs/transformers/v5.19.0/model_doc/glm_moe_dsa?utm_source=chatgpt.com

[16] 000 001 TEACHING PRETRAINED LANGUAGE MODELS TO THI
https://openreview.net/pdf/e443bfddbba382654e298cb5e0f53c2960eb470e.pdf?utm_source=chatgpt.com

19 GLM-5.1: Towards Long-Horizon Tasks
https://z.ai/blog/glm-5.1?utm_source=chatgpt.com

20 LiveCodeBench Leaderboard and Methodology | Vals AI
https://www.vals.ai/benchmarks/lcb?utm_source=chatgpt.com

21 常见问题解答 (Q&A) | ZCode Docs
https://zcode.z.ai/cn/docs/qa?utm_source=chatgpt.com

22 24 连接模型 | ZCode Docs
https://zcode.z.ai/cn/docs/configuration?utm_source=chatgpt.com

25 GLM-5/README.md at main · zai-org/GLM-5 · GitHub
https://github.com/zai-org/glm-5/blob/main/README.md?utm_source=chatgpt.com

28 GLM-5: From Vibe Coding to Agentic Engineering
https://z.ai/blog/glm-5?trk=public_profile__posts-text&utm_source=chatgpt.com