実用的観点から見た CoT 蒸留における教師–生徒能力差の影響

梶塚 時央 [1]　本多 右京 [2]　高瀬 翔 [2]
[1] 東京大学大学院　[2] サイバーエージェント
kajitsuka-tokio@g.ecc.u-tokyo.ac.jp　{honda_ukyo, takase_sho}@cyberagent.co.jp

概要

Chain-of-thought（CoT）蒸留は，教師モデルの推論を小型モデルに転移して大規模言語モデルの運用コストを下げる．一方，教師と生徒の能力差が大きいと蒸留が不調になる現象が報告されており，適切なモデル選択にはコストがかかる．本稿は既存の実験設定を実運用に合わせて見直すことで，この現象の実際的な影響を再検討する．まず，既存設定では蒸留後にむしろ性能が低下していることを示す．次にこれを是正した設定で再評価を行い，能力差の影響はタスクや設定を通じて一貫して見られるものではなく，特に候補教師間の性能差が大きい場合は見られないことを示す．以上を通じて，教師・生徒ペア選択と評価に関する実用的指針を提示する．

1　はじめに

大規模言語モデル（LLM）は，最終解だけでなく **Chain-of-Thought (CoT)** と呼ばれる推論過程を生成させることで，高い推論性能を示すことが知られている [1, 2]．しかし，これにより出力が長くなり，推論コストが大きく増加してしまう．この課題に対し，より小型で効率的な生徒モデルに教師モデルの推論を模倣させる **CoT 蒸留**が提案されてきた [3, 4, 5, 6, 7, 8]．CoT 蒸留の詳細は付録 A を参照．

CoT 蒸留における重要な懸念として，教師と生徒間の能力差（**Capacity Gap**）の問題がある．既存研究は，教師と生徒の能力差が大きいと蒸留による性能の改善幅が小さくなることがあると報告している [9, 10]．つまり，最高性能の教師を使うことが最適な蒸留戦略とはならない場合があり，生徒に見合う教師を探索する必要が生じる．しかし，この探索は蒸留の学習・評価コストを押し上げてしまう．

本稿はこの能力差の影響を実用的観点から検証する．検証により，まず既存設定が現実の蒸留運用を十分に反映していないことを示す．具体的には，蒸留後モデルが蒸留前の性能を下回っていること，他の教師も正答した事例のみ教師データとして使うフィルタリングや，生徒が教師より大きいという非現実的な設定の混入が問題としてある．

これらを踏まえ，より現実的な実験設定での再検証を行う．結果として，教師と生徒の能力差の影響はタスクや設定によっては見られず，特に候補教師の性能差が大きい場合には高性能教師の利点が能力差の影響を上回ることが多いと分かった．以上から実用的指針として，(1) 蒸留後に性能が改善することを確認すること，(2) 候補教師間の性能差が大きい場合は高性能教師を優先すること，を提案する．

2　既存設定の問題点

本節では，CoT 蒸留での能力差の影響を報告した先行研究 [9, 10] における実験設定を実用的観点から再検討する．問題点は以下の 3 点である．

2.1　蒸留前ベースラインとの比較の欠如

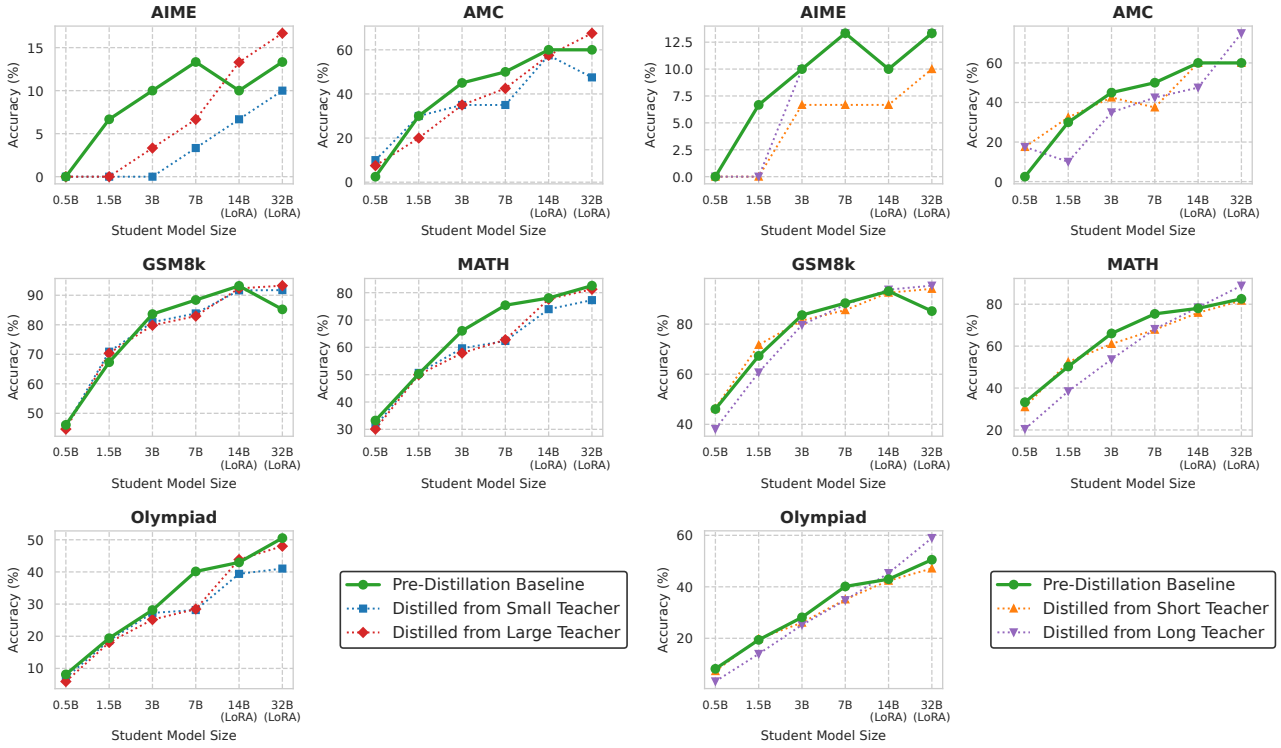
先行研究 [9, 10] は蒸留後のモデル同士の比較に留まり，蒸留前の性能を上回るかの検証を行っていない．しかし，LLM はゼロショットまたは少数例示で高い推論能力を示すことがあり，蒸留がこれに対して性能改善をもたらすことは自明ではない．そこで，まず蒸留による性能改善の確認を行う．

実験設定．　先行研究 [9] の公開コードに基づき，蒸留前モデルを追加して比較を行った．モデル能力差の検証は教師モデルのサイズ（**Small–Large**）と教師の CoT の長さ（**Short–Long**）の 2 つの観点で行われる．実験設定の詳細は付録 B に示す．

結果．　図 1a および 1b に示すとおり，多くの設定で CoT 蒸留後に蒸留前モデルを**下回る**結果となった．つまり，当該実験設定は，蒸留が有益であるという実運用上の前提を満たしていない．

2.2　教師間でのデータフィルタリング

先行研究 [9] では，比較する教師モデルのペア（Small–Large と Short–Long）において両方の教師が



(a) **Small–Large** 設定における結果.　　(b) **Short–Long** 設定における結果.

図 1: 既存の実験設定での，5 つの数学タスクにおける蒸留前モデルと蒸留後モデルの性能比較.

正答した事例の CoT のみ蒸留に使用するというフィルタリングを行う．これは，純粋なモデル能力差の影響を切り出して検証するという意味では合理的である．教師間で学習データ量を等しくすることで，蒸留の性能差を CoT の質に帰属しやすくなる．

しかし，実運用では教師が正答した事例の CoT を最大限活用したいはずであり，このフィルタは現実的ではない．特に，強い教師の利点である，正しく正解に至る CoT 例数の増加と，難例を含む分布カバーの拡大を同時に打ち消してしまう．

2.3　教師より生徒が大きい設定の混入

生徒が教師より大きい設定 [9] は，能力差の包括的な検証という観点では興味深いが，推論コスト削減を目的とする CoT 蒸留の動機とは整合しない．実用的に考慮すべき設定は，生徒に対して教師の能力が高いものに限定される．

3　実用的な設定での再評価

前節の問題を踏まえ，本節では，実運用に近い実験設定において教師–生徒能力差の影響を検証する．

3.1　実験設定の修正

蒸留の有効性が期待されるタスク選定．　近年の LLM でも事前知識だけでは解けず，学習が必要となるタスクを選定する．**BIG-Bench Hard (BBH)** [11] は GPT-4.1 に対しても文脈内での学習事例の追加で性能向上があり [3]，有望な候補と考えた．さらに，BBH 内の各タスクで教師群と生徒群の少数例示学習（few-shot in-context learning）での性能を測定し，2 群の平均差が大きい 15 タスクを，学習余地が大きく蒸留の効果が期待されるタスクとして選定した．

教師間データフィルタリングの撤廃．　各教師について，当該教師が正答した例の CoT をすべて蒸留に用いる．これにより，教師–生徒能力差のみの影響を分離して測ることは難しくなるが，強い教師がより多く，より難しい学習事例を提供できるという実運用上の利点を含めた総合効果を評価できる．

生徒モデルサイズへの制約．　生徒が教師より厳密に小さい設定に限定し，運用コスト削減という蒸留の目的に整合させる．

上記 3 点の修正以外は 2.1 節の設定にしたがう．その他の詳細は付録 C に示す．

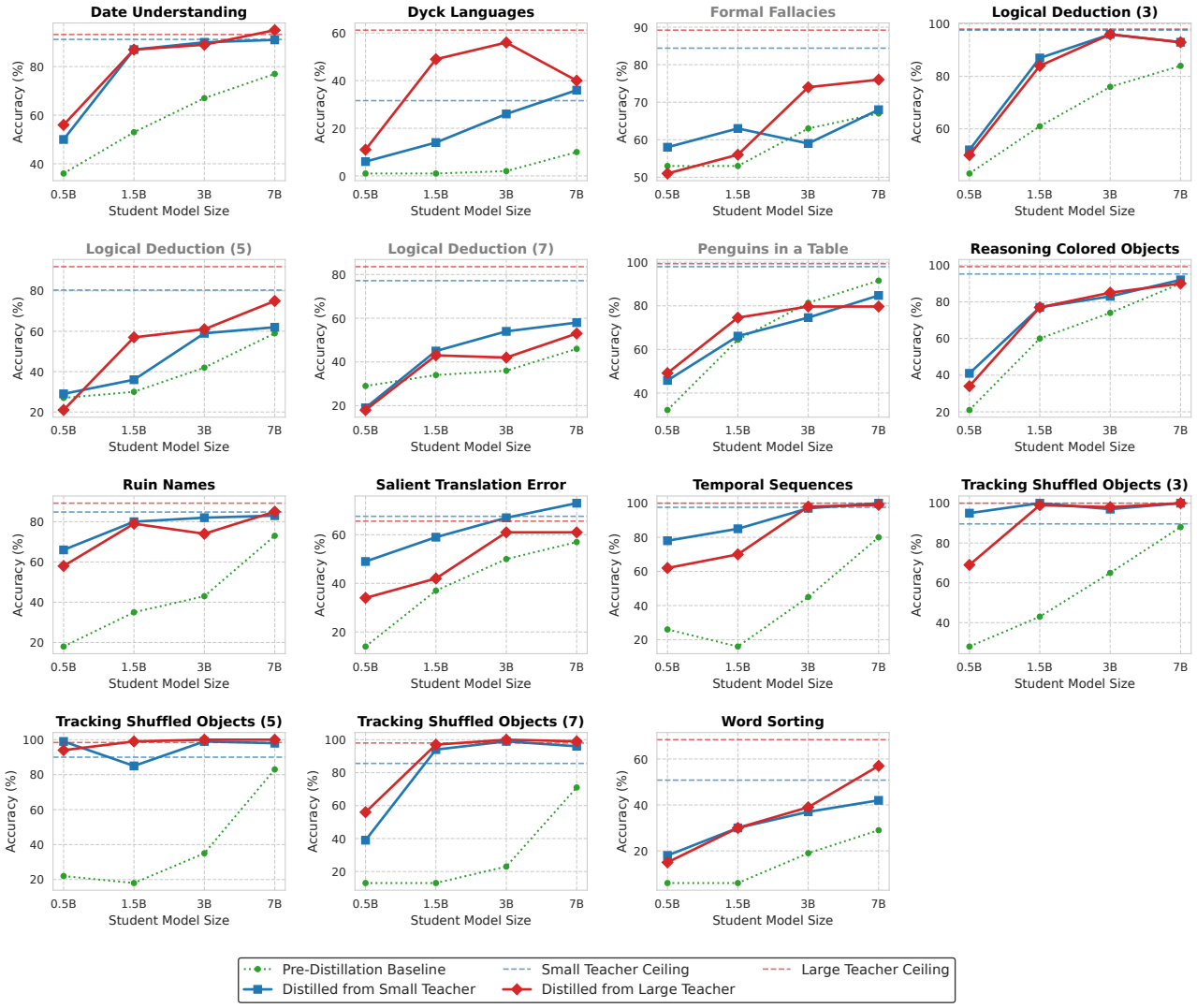


図 2: BBH タスクにおける **Small–Large** 設定の結果．灰色のタスク名では蒸留後の性能低下が見られた．

3.2 結果

図 2 および 3 に結果を示す．

蒸留の有効性． 多くのタスクで蒸留が有効だったが，Formal Fallacies，Logical Deduction，Penguins in a Table では一部サイズで性能低下が見られた．以降は，蒸留で一貫して改善したタスク（Small–Large では 11，Short–Long では 10）を主に分析する．

教師–生徒能力差の影響． 修正後の設定では，一貫した能力差の影響は見られない．Small–Large では，蒸留が有効な 11 タスク中 8 タスクで能力差の影響（小生徒では弱教師が有利だが生徒が大きくなると差が縮むまたは反転する）が観測された一方，残りの 3 タスク（Date Understanding，Dyck Languages，Tracking Shuffled Objects (7)）では観測されなかった．Short–Long では，Temporal Sequences と Tracking Shuffled Objects (7) のみで同様の傾向が見られ，残り 8 タスクでは見られなかった．

教師間能力差の影響． 一方，Dyck Languages や Word Sorting のように候補教師間の性能差が大きいタスクでは，生徒サイズに依らず強い教師が同等以上の生徒を得ることが多い．すなわち，教師と生徒の能力差が存在していても，教師性能差が大きい状況では強い教師の利点が上回りやすい．

蒸留が有効でないタスクの分析． 蒸留効果が限定的だった 3 タスクでは，生徒と教師間，教師間の能力差のいずれも，一貫した影響を示さなかった．

4 考察

本節では結果を実用上の指針として整理する．

指針 1：蒸留前ベースラインに対する蒸留の有効性を必ず確認する． 近年の LLM は微調整なしで

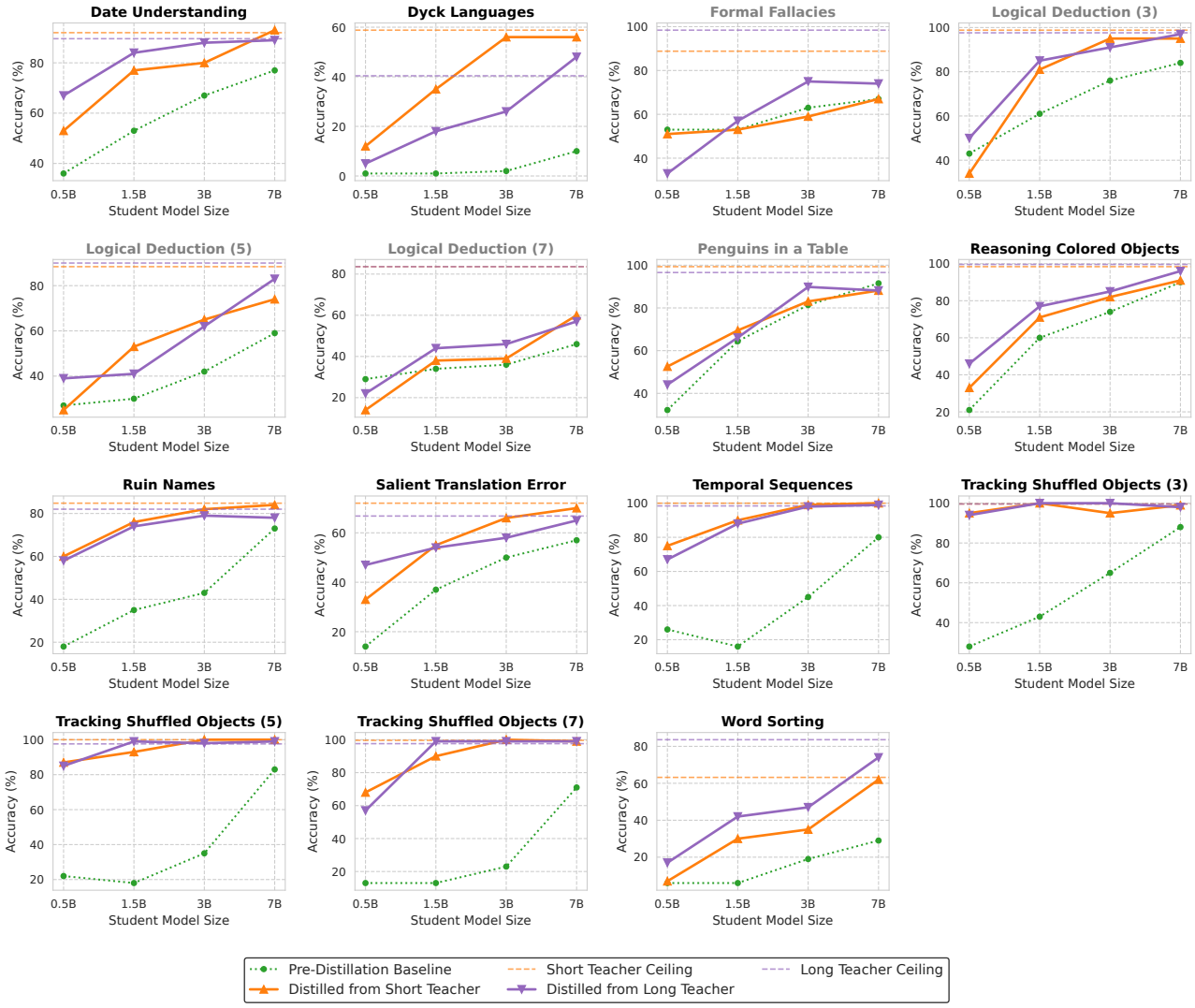


図 3: BBH タスクにおける **Short–Long** 設定の結果．灰色のタスク名では蒸留後の性能低下が見られた．

も高い推論能力を持つため，蒸留が常に有効とは限らない．運用投入や蒸留戦略の比較の前に，蒸留が性能を改善しているかを確認すべきである．今回の基準で選定した BBH ではほとんどの生徒で蒸留が有効であったことから，少数例示学習での性能差が，蒸留の適否を判断するための軽量な指標として利用できる可能性がある．

指針 2：教師性能差が大きい場合は，高性能な教師を優先する． 蒸留が有効で，かつ候補教師間に大きな性能差がある場合は，高性能な教師を優先するのが妥当である．実用的な設定では強い教師はより多くの正答例を提供でき，このデータ量利得が教師と生徒の能力差の影響を上回ると考えられる．本実験では，教師間で提供例数が概ね 1.3 倍以上異なるタスクでこの傾向が顕著だった．性能差が小さい場合は，教師の推論コストや CoT 長（短い CoT は推論コストを下げる）も含めて考慮する必要がある．

5 おわりに

本稿では，CoT 蒸留における生徒と教師間の能力差の影響を実用観点から再評価した．この結果，既存の実験設定では蒸留がむしろ性能を低下させていることや，教師の活用や推論効率改善の観点から非現実的な設定になっていることを明らかにした．修正後設定での再評価により，能力差の影響はタスクや設定を通じて見られるものではなく，候補教師間の性能差が大きい場合は生徒のサイズによらず強い教師が有効であることを示した．実際に蒸留を用いる際には，(1) 蒸留が有効なタスクであるか確認する，(2) 教師間性能差が大きいなら高性能教師を優先する，ことを推奨する．今後の課題として，より幅広いタスクやモデルでの検証が挙げられる．

参考文献

[1] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, **NeurIPS**, Vol. 35, pp. 24824–24837. Curran Associates, Inc., 2022.

[2] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, **NeurIPS**, Vol. 35, pp. 22199–22213. Curran Associates, Inc., 2022.

[3] Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **ACL**, pp. 14852–14882, Toronto, Canada, July 2023. Association for Computational Linguistics.

[4] Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **ACL**, pp. 2665–2679, Toronto, Canada, July 2023. Association for Computational Linguistics.

[5] Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. Specializing smaller language models towards multi-step reasoning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, **ICML**, Vol. 202 of **Proceedings of Machine Learning Research**, pp. 10421–10430. PMLR, 23–29 Jul 2023.

[6] Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Findings of ACL 2023**, pp. 8003–8017, Toronto, Canada, July 2023. Association for Computational Linguistics.

[7] Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **ACL**, pp. 1773–1781, Toronto, Canada, July 2023. Association for Computational Linguistics.

[8] Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. Distilling reasoning capabilities into smaller language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Findings of ACL 2023**, pp. 7059–7073, Toronto, Canada, July 2023. Association for Computational Linguistics.

[9] Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Findings of ACL 2025**, pp. 25366–25394, Vienna, Austria, July 2025. Association for Computational Linguistics.

[10] Xinghao Chen, Zhijing Sun, Guo Wenjin, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, and Xiaoyu Shen. Unveiling the key factors for distilling chain-of-thought reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Findings of ACL 2025**, pp. 15094–15119, Vienna, Austria, July 2025. Association for Computational Linguistics.

[11] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Findings of ACL 2023**, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics.

[12] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In J. Vanschoren and S. Yeung, editors, **NeurIPS Track on Datasets and Benchmarks**, Vol. 1, 2021.

[13] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In Yixin Cao, Yang Feng, and Deyi Xiong, editors, **ACL**, pp. 400–410, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[14] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. **arXiv preprint arXiv:2110.14168v2**, 2021.

[15] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **ACL**, pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[16] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024.

表 1: BBH タスクとその概要.

タスク	概要
Date Understanding	文脈から日付を読み取り，形式変換や期間計算などに回答する.
Dyck Languages	括弧列の不足している閉じ括弧を補完する.
Formal Fallacies	非形式的な議論の論理的妥当性を判定する.
Logical Deduction 3/5/7	3/5/7 個の物体の相対位置に関する手がかりから，ある物体の正しい位置を答える.
Penguins in a Table	ペンギンの属性をまとめた表から，条件に該当するペンギンやその数などを答える.
Reasoning Colored Objects	物体の配置を記述した文章から，指定された物体の色（机・棚などの上のどれが何色か）を答える.
Ruin Names	与えられたアーティスト名・バンド名・映画名を 1 文字だけ改変し，ユーモラスな名称を生成する.
Salient Translation Error	ドイツ語原文と誤りを含む英訳を比較し，翻訳エラーの種類（固有名詞・数値・否定など）を分類する.
Temporal Sequences	人物の日次スケジュールから，特定の出来事が起こった時間帯を答える.
Tracking Shuffled Objects 3/5/7	3/5/7 個の物体に対する交換操作を追跡し，最終的な所有者を答える.
Word Sorting	単語をアルファベット順に並べ替える.

A CoT 蒸留

CoT 蒸留は，大規模な教師モデルの推論過程を，小型の生徒モデルへ移すことを目的とする．典型的には次の 2 段階で進む.

推論過程（CoT）の生成． 教師モデル $\mathcal{T}$ は，訓練データ $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ に対し，推論過程 $\{\hat{r}_i\}$ と予測解 $\{\hat{y}_i\}$ を生成する．推論過程の品質を担保するため，予測解が正解と一致した例のみを残す.

$$\mathcal{D}_{\mathrm{CoT}} = \{(x_i, \hat{r}_i, \hat{y}_i) \mid (x_i, y_i) \in \mathcal{D},\ \hat{y}_i = y_i\}. \quad (1)$$

生徒モデルの微調整． 生徒モデル $\mathcal{S}$ は $\mathcal{D}_{\mathrm{CoT}}$ 上で以下の負の対数尤度を最小化するよう学習する.

$$\mathcal{L} = -\sum_{(x,\hat{r},\hat{y})\in\mathcal{D}_{\mathrm{CoT}}} \log P_{\mathcal{S}}(\hat{r}, \hat{y} \mid x). \quad (2)$$

B 既存実験設定詳細

学習・評価ともに，実験は先行研究 [9] の公開コードに基づいて行った.[1] 蒸留データも著者らによって公開されている．MATH の学習セット [12] から構成され，モデルサイズの異なる教師を比べる Small–Large（Qwen2.5-3B-Instruct vs Qwen2.5-72B-Instruct）設定と，推論長の異なる教師を比べる Short–Long（Qwen2.5-32B-Instruct vs QwQ-32B-Preview）設定それぞれで作成されている．生徒には Qwen2.5-{0.5B, 1.5B, 3B, 7B, 14B, 32B}-Instruct を用い，`LLaMA-Factory` [13] で先行研究と同一ハイパーパラメータで微調整した．評価には難易度の異なる 5 つの数学ベンチマーク（MATH，GSM8K [14]，AMC 2023，AIME 2024，OlympiadBench [15]）を用いた．蒸留前ベースラインを含む全モデルに，先行研究と同一のプロンプトを適用した.

1） https://github.com/Small-Model-Gap/Small-Model-Learnability-Gap

C 修正実験設定詳細

基本的な実験設定は 2.1 節（付録 B）と同じである．BBH でのタスク選別では，各タスクにおいて教師と生徒の少数例示学習での精度を測定し，教師群と生徒群の平均差が 30 ポイントを超えるものを，学習余地が大きく蒸留の効果が期待されるタスクとして選定した．表 1 に，これにより得られた 15 タスクの概要を示す．各タスクは 3:2 で学習・テストに分割する．教師間のデータフィルタリングは行わず，各教師が正解した事例とその CoT をその教師の蒸留データとする．生徒モデルの数を確保しつつサイズを教師より小さく保つため，生徒には Qwen2.5-{0.5B, 1.5B, 3B, 7B}-Instruct を用い，Small–Large 設定では Qwen2.5-14B-Instruct と Qwen2.5-72B-Instruct をそれぞれ小・大教師として用いた．Short–Long 設定では，2.1 節と同じく，Qwen2.5-32B-Instruct と QwQ-32B-Preview を用いる．評価には `lm-evaluation-harness` [16] を用い，蒸留前モデルを含むすべてのモデルで，同じプロンプト（CoT 付きの少数例示）を適用して推論を行った.