

Efficient Chain-of-Thought Distillation for Compact Language Models under Limited Computational Budgets

Custodio Corominas

Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS, USA.

custodiocorominas748@ku.edu

Vinzenz Mitschke

Department of Computer Science, George Mason University, Fairfax, VA, USA.

vinzenzmitschke189@gmu.edu

Ashlee Costa

Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, USA.

ashleecosta@buffalo.edu

Majvor Alm

School of Information Technology, University of Cincinnati, Cincinnati, OH, USA.

almmajvor@uc.edu

Abstract

Chain-of-thought prompting has become a central technique for eliciting multi-step reasoning in large language models, but its computational demands make direct deployment impractical for many organizations, edge environments, and budget-constrained research settings. This paper examines efficient chain-of-thought distillation for compact language models under limited computational budgets. Rather than treating distillation as a purely algorithmic compression problem, the analysis frames it as a socio-technical systems challenge involving teacher trajectory generation, data filtering, student capacity allocation, training infrastructure, evaluation, governance, and lifecycle sustainability. The paper argues that effective distillation depends less on transferring verbose rationales verbatim than on selecting, filtering, and aligning reasoning traces so that compact students acquire transferable procedural competence. It further examines structural trade-offs among teacher scale, trajectory quality, student architecture, quantization, adapter-based tuning, inference latency, and evaluation rigor. Particular attention is given to robustness, fairness, and policy implications, because compact reasoning models may be deployed in high-stakes settings where opaque failure modes and biased supervision can produce uneven harm. The paper concludes that efficient chain-of-thought distillation should be designed as a governed pipeline rather than a single training run, with explicit attention to data provenance, evaluation validity, deployment constraints, and environmental cost. Such a pipeline can widen access to reasoning-capable models while preserving accountability, but only if efficiency gains are matched by system-level transparency and continuous monitoring.

Keywords

chain-of-thought distillation; compact language models; computational budgets; reasoning transfer; systems architecture; sustainability; fairness; governance.

1. Introduction

Large language models have demonstrated substantial reasoning abilities when prompted to produce intermediate steps before final answers, a paradigm commonly associated with chain-of-thought prompting [1]. This approach can improve performance on arithmetic, symbolic, commonsense, and multi-hop tasks by encouraging models to externalize intermediate computations. However, the same property that makes chain-of-thought useful also makes it costly. Longer outputs increase inference latency, memory pressure, and energy consumption, while the largest teacher models often require hardware that is unavailable to smaller laboratories, public institutions, and edge deployments. Zero-shot prompting methods have reduced some dependence on manually crafted exemplars, yet they do not eliminate the fundamental cost of invoking very large models for every reasoning request [2].

Compact language models offer a pragmatic alternative. They can run on modest accelerators, support on-device inference, and reduce operational expenditure. Their weakness is that they often fail to reproduce the multi-step coherence of larger teachers, especially when tasks require compositional reasoning or robust handling of distractors. Chain-of-thought distillation seeks to close this gap by transferring reasoning behavior from a capable teacher to a smaller student. The challenge is not simply to copy rationales. A student with fewer parameters may memorize surface patterns without internalizing the underlying procedure, and a teacher may generate fluent but invalid reasoning paths. Under limited computational budgets, the system designer must decide where to spend scarce resources: on teacher inference, trajectory filtering, student training, evaluation, or deployment optimization.

This paper addresses that decision problem from a system-level perspective. It argues that efficient chain-of-thought distillation should be understood as an infrastructure and governance problem, not merely as a model compression technique. The discussion covers conceptual foundations, teacher trajectory generation, student architecture, efficiency trade-offs, robustness and fairness, deployment and policy, and sustainability. The central claim is that compact reasoning models can be made more reliable and accessible when distillation pipelines are designed around filtered, aligned, and auditable reasoning traces rather than around raw teacher output volume. Efficiency, in this view, is not only a matter of reducing floating-point operations. It is also a matter of reducing wasted supervision, invalid evaluation, operational fragility, and long-term maintenance burden.

2. Conceptual Foundations and System-Level Problem Framing

Knowledge distillation traditionally involves training a smaller student to mimic the outputs or internal representations of a larger teacher [3]. In its classical form, distillation compresses predictive competence by matching softened probability distributions or intermediate features. Compact models such as DistilBERT and TinyBERT demonstrated that smaller architectures can retain much of the performance of larger transformers on natural language understanding tasks when trained with careful distillation objectives [4][5]. Surveys of knowledge distillation have since mapped a wide design space involving response-based, feature-based, and relation-based transfer [6]. Chain-of-thought distillation extends this space by treating intermediate reasoning steps as part of the supervisory signal. The student is expected not only to produce the correct final answer but also to generate a coherent path that justifies or explains it.

This extension introduces distinct system-level problems. First, reasoning traces are sequential and often long, which raises the cost of storing, filtering, and training on them. Second, the quality of a trace is not identical to the correctness of its final answer. A model may reach a correct answer through flawed reasoning, or a correct reasoning path may be expressed in a style that a compact student cannot imitate. Third, the teacher may exhibit biases, verbosity, or inconsistency across prompts. If these properties are transferred uncritically, the student may inherit undesirable behavior while appearing to improve on benchmark accuracy.

From a structural perspective, chain-of-thought distillation involves several coupled layers. The teacher layer produces candidate trajectories under particular prompts and decoding settings. The data layer filters, deduplicates, labels, and stores those trajectories. The student layer determines model capacity, tokenization, context length, and adaptation method. The training layer manages batching, gradient accumulation, mixed precision, and checkpoint selection. The deployment layer controls latency, memory, privacy, and updates. The governance layer defines who may use the model, for what purposes, and with what oversight. Efficiency under limited computational budgets emerges from the interaction of these layers, not from any single optimization. A cheap teacher may generate noisy traces that increase downstream training cost, while an expensive teacher may produce high-quality traces that reduce the number of student iterations required. Similarly, a smaller student may lower inference cost but require more careful curriculum design to avoid brittle reasoning.

3. Teacher Trajectory Generation and Data Infrastructure

Teacher trajectory generation is the first major cost center in chain-of-thought distillation. A capable teacher can be prompted to produce step-by-step solutions for a large corpus of problems, but doing so at scale requires substantial inference compute. Under limited budgets, indiscriminate generation is wasteful. The system should prioritize examples that are informative, diverse, and aligned with the target deployment distribution. Adaptive trajectory filtering and alignment methods can reduce the volume of low-value reasoning paths by selecting those that are consistent, concise, and pedagogically useful for small parameter models [7]. This suggests that the teacher should not be treated as an oracle whose every output deserves equal weight. Instead, the teacher functions as a proposer, while the pipeline applies filtering, ranking, and sometimes correction.

The quality of teacher traces depends heavily on prompting strategy, decoding temperature, and answer verification. Self-consistency methods show that sampling multiple reasoning paths and selecting the most consistent answer can improve reliability [11]. In a distillation setting, this property can be used to generate candidate traces and then retain only those that converge on verified answers. However, self-consistency multiplies teacher inference cost, so budget-constrained systems must decide when multiple samples are worth the expense. For tasks with cheap automatic verification, such as arithmetic or code execution, multi-sample filtering may be highly effective. For open-ended tasks with subjective answers, verification becomes more difficult, and human review or model-based critique may be required.

Large language models can act as reasoning teachers even when their final answers are imperfect, provided that the generated rationales are filtered and adapted to student capacity [8]. Teaching smaller models to reason therefore involves a curriculum problem: the student should encounter traces that are complex enough to be informative but not so complex that they exceed the student’s representational capacity. Research on teaching small language models to reason indicates that distilled rationales can improve performance on multi-step

tasks, but the benefits depend on task selection, model size, and training regime [9]. The data infrastructure must therefore support metadata about provenance, difficulty, answer verification, and filtering decisions. Without such metadata, later audits cannot determine whether a student’s failure arises from teacher error, data contamination, distribution shift, or architectural limitation.

Data governance is also relevant at this stage. Teacher-generated traces may contain copyrighted text, personal information, toxic content, or biased reasoning. If these traces are stored and reused, they become part of the student’s developmental history. Organizations should define retention policies, access controls, and redaction procedures before large-scale generation begins. In budget-limited settings, there is a temptation to skip such controls because they appear to add overhead. Yet the cost of remediation after a biased or privacy-violating model is deployed can exceed the initial savings. Efficient distillation therefore includes efficient governance: collecting only what is necessary, filtering what is harmful, and documenting what remains.

4. Student Architecture, Capacity Allocation, and Training Regimes

The student architecture determines how much of the teacher’s reasoning behavior can be absorbed. Parameter count is important, but it is not the only factor. Context length, attention pattern, positional encoding, tokenizer compatibility, and depth-to-width ratio all influence reasoning performance. A compact model with a small context window may struggle to represent long chains of thought, while a model with insufficient depth may fail to compose multiple inferential steps. Research on specializing smaller language models for multi-step reasoning shows that targeted training can improve performance, but specialization may also reduce generality [10]. System designers must therefore decide whether the student should be a general-purpose compact model or a domain-specialized reasoner.

Training regimes for chain-of-thought distillation include supervised fine-tuning on teacher traces, preference optimization, and iterative bootstrapping. Self-consistency and bootstrapping methods demonstrate that models can improve by generating reasoning paths and learning from those that lead to correct answers [11][12]. In compact models, iterative bootstrapping can be attractive because it reduces dependence on a large external teacher after initial data generation. However, it also risks reinforcement of the student’s existing errors. If the student generates flawed reasoning and the verifier is weak, the training loop may amplify bias rather than correct it. A robust pipeline therefore combines external teacher supervision with internal self-training, using verification and diversity constraints to prevent collapse.

Capacity allocation is a central trade-off. A larger student may learn more accurate reasoning but cost more to train and deploy. A smaller student may be cheaper but require more extensive data curation. Intermediate approaches include adapter-based tuning, quantized training, and modular architectures. For compact models, parameter-efficient methods can reduce the memory required for adaptation, but they may not fully compensate for limited base capacity. The architectural choice should be tied to the deployment context. A mobile assistant may prioritize latency and memory, while an on-premises analytics engine may prioritize accuracy and auditability. In both cases, the student should be evaluated not only on final-answer accuracy but also on reasoning faithfulness, calibration, and robustness to adversarial inputs.

5. Efficiency Trade-Offs under Limited Computational Budgets

Efficiency in chain-of-thought distillation is often framed as reducing training and inference costs, but the relevant budget includes human labor, data storage, evaluation time, and operational risk. A distillation pipeline that saves graphics processing unit hours but produces an unreliable model may impose larger downstream costs through error correction, user distrust, or regulatory scrutiny. Conversely, a pipeline that spends more on trajectory filtering may reduce the need for repeated fine-tuning and extensive post-deployment monitoring. The optimal allocation depends on the cost of errors in the target domain.

Distilling reasoning capabilities into smaller models has been shown to be feasible, but the gains vary by task and model family [13]. Some tasks benefit from explicit rationales, while others benefit more from answer-only supervision. Progressive learning from complex explanation traces, as explored in instruction-tuning research, suggests that curriculum and data quality can matter as much as raw model scale [14]. For compact models, this implies that a smaller, well-curated dataset of high-quality reasoning traces may outperform a much larger dataset of noisy traces. The system designer should therefore treat data filtering as a first-class efficiency lever rather than as a preprocessing afterthought.

Model compression techniques further shape the budget. Open foundation models have made it easier to obtain strong base models for distillation, but their license terms, tokenizer choices, and training data provenance must be considered [15]. Quantization can reduce memory and inference cost, and methods such as QLoRA enable fine-tuning of quantized models with lower hardware requirements [16]. Low-rank adaptation can also reduce the number of trainable parameters, making domain adaptation more accessible [17]. These techniques are valuable under limited budgets, but they introduce trade-offs in numerical stability, latency, and accuracy. Quantized students may be more sensitive to distribution shift, and adapter-based updates may forget reasoning behavior learned during initial distillation. The pipeline should therefore include regression testing after every efficiency optimization.

Inference efficiency is especially important for chain-of-thought models because reasoning traces increase output length. A compact model that produces long rationales may have higher latency than a larger model that produces short answers, particularly in real-time applications. System designers can manage this trade-off through early exiting, adaptive reasoning depth, caching, or routing simple queries to non-reasoning modes. Such strategies are not purely technical; they determine when a user receives a reasoned answer and when they receive a fast approximation. Governance should specify whether users are informed about these mode switches, because hidden changes in reasoning depth can affect trust and accountability.

6. Robustness, Fairness, and Evaluation in Compact Reasoning Systems

Compact reasoning models are often deployed in resource-constrained settings where continuous human oversight is limited. This makes robustness and fairness central design concerns rather than optional evaluation criteria. A distilled student may inherit the teacher’s vulnerabilities, including susceptibility to prompt injection, spurious correlations, and socially biased reasoning. If the teacher’s traces are filtered only for final-answer correctness, biased or harmful intermediate steps may remain. Evaluation should therefore examine reasoning traces, not just outputs, and should include adversarial, counterfactual, and distribution-shift tests.

Fairness in chain-of-thought distillation has both data and model dimensions. On the data side, teacher-generated traces may underrepresent certain dialects, cultural contexts, or problem framings. On the model side, compact students may have less capacity to represent nuanced

social context, leading to uneven performance across groups. Energy and policy considerations in natural language processing also remind us that efficiency claims must be evaluated against the broader environmental and social costs of model development [18]. A model that is computationally cheap but socially harmful is not efficient in any meaningful system-level sense.

Evaluation methodology must be designed with the same care as training. Final-answer accuracy alone can hide reasoning failures, and benchmark contamination can inflate apparent gains. Model cards provide a structured way to document intended use, performance characteristics, limitations, and ethical considerations [20]. For compact reasoning models, documentation should include the teacher model, trajectory generation date, filtering criteria, verification methods, known failure modes, and deployment constraints. Without such documentation, downstream users cannot assess whether the model is appropriate for their context. The dangers of large-scale language models also underscore the need to consider who benefits from deployment and who bears the risk [19]. Compact models may democratize access, but they can also distribute unaccountable reasoning systems more widely.

7. Deployment, Governance, and Policy Implications

Deployment transforms a distilled model from a research artifact into an operational system. In edge or on-premises settings, the model may be updated infrequently, which increases the importance of initial validation and fallback procedures. In cloud settings, the model may be part of a larger service pipeline that includes retrieval, verification, and human review. Governance should specify accountability for reasoning failures. If a compact model produces an incorrect medical, legal, or financial recommendation, it is insufficient to blame the student model alone. The teacher data, filtering rules, training objectives, deployment configuration, and monitoring practices all contribute to the outcome.

Policy implications extend beyond individual deployments. Funding agencies and regulators may encourage compact reasoning models as a way to reduce compute concentration, but efficiency alone does not guarantee equity. Without transparency requirements, organizations could deploy low-cost reasoning systems in under-resourced communities while reserving larger, more reliable models for wealthy users. Model documentation and audit trails can help counter this tendency [20]. Policies might require disclosure of reasoning model provenance, evaluation coverage, and known limitations, especially in public-sector applications. They might also support shared infrastructure for trajectory generation and verification, so that smaller institutions do not bear the full cost of teacher inference.

Governance must also address update cycles. A compact model that is distilled once may become stale as language, knowledge, and social norms change. Continuous monitoring can detect drift, but it adds operational cost. A practical approach is to define trigger conditions for re-distillation, such as a decline in verification pass rates, an increase in user appeals, or a significant shift in input distribution. These triggers should be documented in advance and tied to responsible parties. In this sense, efficient distillation is not a one-time compression event but a lifecycle process with governance checkpoints.

8. Sustainability and Lifecycle Considerations

Sustainability in chain-of-thought distillation involves both training and inference. Teacher inference can dominate the initial carbon footprint if large models are queried extensively. Student training adds further cost, especially when multiple iterations are required. Inference then accumulates over the deployment lifetime, and long reasoning traces can increase energy

use per query. A lifecycle assessment should compare the total emissions of distillation against the alternative of directly using a larger teacher. In some cases, distillation may reduce total energy use by enabling local inference and avoiding repeated calls to a remote large model. In other cases, the upfront teacher cost may be difficult to offset if the compact student is deployed infrequently.

Hardware choices matter as well. Quantized models and adapters can run on older or smaller accelerators, extending hardware lifetimes and reducing the demand for new specialized chips [16][17]. This can be a significant sustainability benefit, particularly for organizations that cannot afford frequent hardware refresh cycles. However, quantization and adapter methods can also lead to performance regressions that require additional evaluation, which consumes energy and human time. The system designer should therefore measure not only energy per inference but also energy per successful task completion. A model that uses less power per token but requires multiple retries may be less sustainable overall.

Data storage and retention also have environmental costs. Large corpora of teacher-generated reasoning traces require storage, backup, and access infrastructure. Filtering and deduplication reduce storage needs, but they require compute. Open documentation practices, such as datasheets and model cards, can help organizations share evaluation results and avoid redundant teacher generation [20]. Shared benchmarks and public trajectory repositories, governed by clear licensing and privacy rules, could reduce duplicated effort across the research community. Sustainability in this domain is therefore intertwined with openness, standardization, and coordination.

9. Conclusion

Efficient chain-of-thought distillation for compact language models under limited computational budgets is best understood as a system design problem. The goal is not merely to compress a teacher into a smaller student, but to construct a pipeline that generates useful reasoning trajectories, filters them for quality and alignment, trains a student within realistic resource constraints, and deploys that student with appropriate safeguards. The analysis in this paper has emphasized that efficiency is multidimensional. It includes compute, memory, latency, human labor, governance overhead, and environmental impact. A narrow focus on parameter count or floating-point operations can obscure these dependencies.

The paper has also argued that reasoning traces are not neutral data. They carry the teacher’s biases, stylistic patterns, and errors, and they can shape the student’s behavior in ways that final-answer accuracy does not reveal. Robust distillation therefore requires verification, diversity, provenance tracking, and evaluation of intermediate reasoning. Fairness and policy considerations are not external constraints on an otherwise technical process. They are constitutive of whether a compact reasoning model should be deployed at all. Model documentation, audit trails, and update triggers provide practical mechanisms for accountability.

Finally, sustainability requires a lifecycle perspective. Teacher generation, student training, quantization, deployment, and monitoring all contribute to the total cost of ownership. Compact models can widen access to reasoning capabilities, but they can also distribute unaccountable systems more widely. The responsible path forward is to treat distillation as a governed infrastructure project, with explicit trade-offs, transparent evaluation, and continuous revision. If efficiency gains are pursued alongside robustness, fairness, and

sustainability, compact chain-of-thought models can become a meaningful part of the broader artificial intelligence ecosystem rather than a fragile shortcut.

References

1. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 24824-24837.
2. Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35, 22199-22213.
3. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
4. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
5. Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling BERT for natural language understanding. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 4163-4174.
6. Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6), 1789-1819.
7. Li, Q. (2026, May). TrajLite PPO: Scalable Reasoning Path Filtering and Alignment for Small Parameter Large Language Models. In *2026 2nd International Conference on Artificial Intelligence and Digital Ethics (ICAIDE)* (pp. 29-32). IEEE.
8. Ho, N., Schmid, L., & Yun, S.-Y. (2022). Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*.
9. Magister, L. C., Mallinson, J., Adamek, J., Malmi, E., & Severyn, A. (2023). Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 1773-1781). Association for Computational Linguistics.
10. Fu, Y., Peng, H., Sabharwal, A., Clark, P., & Khot, T. (2023). Specializing smaller language models towards multi-step reasoning. In *Proceedings of the 40th International Conference on Machine Learning* (pp. 10421-10430). PMLR.
11. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., & Zhou, D. (2023). Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*. OpenReview.net.
12. Zelikman, E., Wu, Y., Mu, J., & Goodman, N. D. (2022). STaR: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35, 15476-15488.
13. Shridhar, K., Stolfo, A., & Sachan, M. (2023). Distilling reasoning capabilities into smaller language models. *Findings of the Association for Computational Linguistics: ACL 2023*, 7059-7073.
14. Mukherjee, S., Mitra, A., Jawahar, G., Agarwal, S., Palangi, H., & Awadallah, A. (2023). Orca: Progressive learning from complex explanation traces of GPT-4. *arXiv preprint arXiv:2306.02707*.

15. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.
16. Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36, 10088-10115.
17. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*. OpenReview.net.
18. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3645-3650). Association for Computational Linguistics.
19. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 610-623). ACM.
20. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (pp. 220-229). ACM.