UDC 004.8 DOI https://doi.org/10.35546/kntu2078-4481.2026.1.28

V. V. BONDAR
Postgraduate Student at the Informational Security
and Computer Engineering Department
Cherkasy State Technological University
ORCID: 0009-0002-3230-5979

INFERENCE-TIME SCALING AS A UNIVERSAL PRINCIPLE IN MACHINE LEARNING: CLASSIFICATION AND CROSS-DOMAIN ANALYSIS

This paper proposes a formal definition and classification of inference-time scaling methods across machine learning. Inference-time scaling – spending additional computation at prediction time to improve output quality – appears throughout the field but has not been unified within a single framework. Methods as varied as ensemble averaging, Monte Carlo tree search, test-time augmentation, chain-of-thought reasoning in large language models, iterative denoising in diffusion models, and ODE solving in flow matching share one structural property: quality improves with additional inference computation, subject to diminishing returns. Unfortunately, these methods are studied by separate communities, and no existing taxonomy spans more than a single domain. We argue that it blocks the sharing and enrichment of methods across different areas of machine learning.

We propose a classification organized by computational topology – the structure of the additional computation performed at prediction time – comprising six types: sequential refinement, parallel sampling with aggregation, tree/graph search, test-time model adaptation, guided generation, and adaptive computation routing. Each method is decomposed into generation, selection, and allocation policies, providing a uniform analytical language.

The analysis identifies four cross-domain invariants: a universally sublinear cost-quality curve, a verifier bottleneck limiting selection-based approaches, a sequential–parallel duality across all domains examined, and method migration between domains once their shared structure is recognized.

Key words: *inference-time scaling, test-time compute, machine learning, generative models, chain-of-thought, diffusion models, flow matching, ensemble methods, Monte Carlo tree search, adaptive computation.*

В. В. БОНДАР
аспірант кафедри інформаційної безпеки та комп'ютерної інженерії
Черкаський державний технологічний університет
ORCID: 0009-0002-3230-5979

МАСШТАБУВАННЯ НА ЕТАПІ ІНФЕРЕНСУ ЯК УНІВЕРСАЛЬНИЙ ПРИНЦИП У МАШИННОМУ НАВЧАННІ: КЛАСИФІКАЦІЯ ТА МІЖДОМЕННИЙ АНАЛІЗ

У роботі запропоновано формальне визначення та узагальнену класифікацію методів масштабування на етапі інференсу в машинному навчанні. Під масштабуванням на етапі інференсу розуміємо практику збільшення обчислювальних витрат під час прогнозування, щоб підвищити якість вихідного результату. Такі підходи трапляються в багатьох підгалузях, проте досі їх не було зведено до спільної системи. До них належать, зокрема, усереднення ансамблів, пошук Монте-Карло по дереву, аугментація на етапі тестування, міркування у форматі "ланцюжку роздумів" у великих мовних моделях і ітеративне розшумлення в дифузійних моделях. Попри різну природу, ці методи об'єднує одна структурна ознака: додаткові обчислення під час інференсу, як правило, покращують якість, однак приріст зменшується зі зростанням обчислювального бюджету. Водночас їх традиційно аналізують всередині окремих задач, а наявні таксономії здебільшого залишаються доменно-специфічними.

Ми вводимо класифікацію за обчислювальною топологією, тобто за тим, як організовано додаткові обчислення на етапі прогнозування. Вона охоплює шість класів: послідовне уточнення, паралельне семплювання з подальшою агрегацією, пошук у дереві або графі, адаптація моделі на етапі тестування, керована генерація та адаптивна маршрутизація обчислень. Для уніфікованого опису кожен підхід подано через три компоненти: стратегія генерації, стратегія відбору та стратегія розподілу обчислювальних ресурсів.

Порівняльний аналіз дозволяє виділити чотири міждоменні інваріанти: загалом сублінійну залежність між додатковою «вартістю» обчислень і приростом якості; вузьке місце моделі оцінювання, що обмежує підходи, побудовані на відборі; послідовно-паралельну дуальність, яка проявляється в усіх розглянутих доменах; перенесення методів між доменами після явного опису їх спільної структури.

Ключові слова: *масштабування на етапі інференсу, обчислення під час тестування, машинне навчання; генеративні моделі; «ланцюжок роздумів»; дифузійні моделі; узгодження потоків; ансамблеві методи; пошук Монте-Карло по дереву; адаптивні обчислення.*

Problem Statement

Scaling computational resources at training time has, for the past decade, served as the dominant path toward better models. Kaplan and colleagues documented power-law dependencies between language model loss and training budget over seven orders of magnitude [1]; Hoffmann and colleagues refined the optimal split between model capacity and data volume for a fixed budget [2]. Following these laws, architectures grew from millions to hundreds of billions of parameters. But the trajectory has limits: training costs rise steeply with each generation while marginal gains narrow, high-quality data is running thin in many domains, and deploying very large models raises latency and infrastructure issues.

A different line of work has been gaining ground: spending extra computation at prediction time rather than at training time. Chain-of-thought prompting has the model write out intermediate reasoning, each token costing a forward pass [3]. Diffusion models treat the denoising step count as a quality dial [4]. Flow matching varies ODE integration precision [5]. Monte Carlo tree search devotes a variable simulation budget to each move in the game [6]. Ensembles aggregate outputs of multiple models at test time [7]. Different as they look, these methods share one feature: more computation at prediction time means better results, with diminishing returns.

The trouble is that these methods inhabit separate literatures. Work on test-time compute for language models does not cite work on denoising schedules; ensemble theory and tree search belong to different scholarly traditions. Surveys published to date cover language models only [8, 9]. On top of that, several well-known techniques – test-time augmentation [10], Monte Carlo Dropout [11], in-context learning [12] – clearly qualify as inference-time scaling but are rarely categorized that way. The result is a fragmented picture that blocks cross-domain transfer and conceals regularities holding across architectures.

In our earlier work we identified parallels between iterative refinement in large language models, diffusion models, and flow matching [13]; the probability transformation function framework provided the mathematical grounding [14]. What the present paper does is extend this analysis from three families of generative models to the whole of machine learning – classical methods, search algorithms, test-time adaptation, and adaptive computation included – to address the absence of a classification covering the full scope of inference-time scaling.

Analysis of Recent Research and Publications

Scaling laws and the training–inference boundary. The scaling laws established by Kaplan and colleagues [1], together with the compute-optimal training ratios proposed by Hoffmann and colleagues [2], both assumed that inference is a fixed-cost operation – a single forward pass per input. Snell and colleagues broke with this assumption. Their experiments showed that an adaptive test-time strategy, one that chooses between sequential chain-of-thought reasoning and parallel sampling with verification depending on query difficulty, can match a model fourteen times larger [15]. That result put inference-time compute on the same footing as model size and training data as a scaling axis.

Large language models. Chain-of-thought prompting, introduced by Wei and colleagues, shows that generating intermediate reasoning steps improves multi-step task performance at proportional computational cost [3]. Self-consistency, proposed by Wang and colleagues, redirects that cost from sequential depth into parallel breadth: multiple reasoning chains are sampled, and the final answer is chosen by majority vote [16]. Yao and colleagues generalized both under a tree-search formulation, with chain-of-thought and self-consistency falling out as limiting cases [17]. How well parallel strategies work depends almost entirely on the quality of the verifier. Process reward models that give per-step feedback outperform outcome-level scoring [18], but even the best learned verifiers saturate: Brown and colleagues found that coverage – the chance of having at least one correct answer in the sample pool – keeps growing as a power law, while the ability to pick that answer out levels off much sooner [19]. A separate development has been the emergence of reasoning models trained via reinforcement learning – OpenAI's o1 [20], DeepSeek-R1 [22], and budget-forcing approaches like s1 [22] – which internalize test-time scaling into the model itself. Hao and colleagues ran MCTS over LLM-generated reasoning, connecting this to classical planning [23].

Diffusion and flow matching. In denoising diffusion models, generation proceeds through T noise-removal steps; quality depends on T but saturates after a point [4]. Song and colleagues unified diffusion with score matching under a continuous SDE framework, turning the step count into a smooth control parameter [24]. A wave of later work – deterministic reverse processes, high-order ODE solvers, restart strategies – reshaped the cost-quality curve without touching the trained weights, each method rethinking how best to spend a given inference budget. Guidance mechanisms add per-step cost in exchange for steered outputs: classifier guidance injects external classifier gradients [25], while classifier-free guidance interpolates conditional and unconditional scores at each step [26]. Ma and colleagues took a different path entirely, spending inference compute on searching over the space of initial noise seeds and selecting the best output through a learned verifier [28] – a procedure that mirrors best-of-N sampling in language models down to its structure. Flow matching learns a velocity field transporting prior samples to data via ODE integration; the number of integration steps directly controls quality [5], and adaptive solvers concentrate steps wherever the dynamics are rigid.

Classical methods and search. Ensemble methods [7], Monte Carlo Dropout [11], and test-time augmentation [10] represent straightforward cases of spending more inference compute to get better predictions. Test-time training [28], entropy-based adaptation via TENT [29], and meta-learning through MAML [30] take a different route, investing test-time

computation in adjusting the model's parameters for each input or task. Beam search sets decoding quality through beam width [31]. The MCTS lineage – AlphaGo and its successors AlphaZero and MuZero [6] – turns simulation budget into playing strength. AlphaFold's recycling mechanism [32] and AlphaCode's million-sample generation pipeline [33] are further domain-specific instances of the same idea.

Existing surveys. Welleck and colleagues [8] and Balachandran and colleagues [9] have both published surveys on inference-time methods, as have several other groups, yet every one of them stays within the language-model domain. Diffusion, flow matching, ensembles, search, test-time adaptation – none of these appear. Our own prior analysis of generative model parallels [13] and the probability transformation function framework [14] forms the starting point that the present paper broadens to machine learning at large.

Research Objective

The goal of this study is to build a classification of inference-time scaling methods covering machine learning in its entirety. More specifically, the work sets out to: give a formal definition of inference-time scaling as a general principle; propose a taxonomy whose primary axis is the computational topology of the extra work done at prediction time; place methods from classical machine learning, search, generative modeling, and adaptive computation within that taxonomy; and draw out cross-domain regularities – in cost scaling, in the limits of verification, and in the interplay of sequential and parallel strategies – that hold across architectures and application areas.

Main Research Material

Formal definition. Take $f_\theta(x)$ to be a base model producing output through a single forward pass at a cost $C_{\min}$. We call any procedure $y^* = \mathrm{ITS}(f_\theta, x, C)$ an inference-time scaling method if the expected output quality $E[Q(y^*)]$ does not decrease as the computational budget C grows beyond $C_{\min}$. The trade-off characteristic of such methods is that cost rises at least linearly with additional work while quality grows sublinearly – typically as $Q \propto \log C$ or $Q \propto C^\alpha$ for some $\alpha < 1$.

Any method fitting this definition breaks down into three parts. A generation policy π_{gen} dictates how additional outputs or refinement steps come into being – one after another, in parallel, or through branching. A selection policy π_{sel} determines how the results are handled: averaging, voting, verifier-based scoring, or convergence to a fixed point. An allocation policy π_{alloc} splits the total budget between generation and selection. The decomposition works across domains without modification. In diffusion sampling, generation is sequential, selection is implicit, allocation is fixed. Self-consistency generates in parallel, selects by vote, and allocates uniformly. MCTS generates along a tree, selects via UCB, and allocates a simulation budget. Test-time augmentation generates in parallel, selects by averaging, and allocates a fixed number of augmentations. One framework fits all four.

Type I: Sequential Refinement. The output passes through a chain of dependent steps, each taking its predecessor's result as input: $x_0 \to f_\theta(x_0) = x_1 \to \ldots \to x_n$. Cost is $O(n \cdot C_{step})$. Under this heading fall the denoising iterations in diffusion models [4], the integration steps in flow matching where finer discretization tightens trajectory accuracy [5], chain-of-thought reasoning in which each token refines the model's working representation of the problem [3], the extended reasoning chains produced by RL-trained systems such as o1 and DeepSeek-R1 [20, 21], the recycling passes in AlphaFold that feed structure predictions back as input [32], and adaptive-depth architectures where the number of processing layers varies per input. What ties these together is causal dependency: each step waits on the one before it, and there is no way around the sequential bottleneck.

Type II: Parallel Sampling with Aggregation. Here, multiple independent outputs are drawn from the same input, then combined or filtered. Cost is $O(n \cdot C_{single})$, and the work is fully parallelizable. The subtypes varies on how aggregation is done. In sample-then-select, an external verifier picks the strongest candidate – reward models for language tasks [18] or unit tests for code generation, as in AlphaCode [33]. In sample-then-vote, majority agreement serves as a stand-in for correctness, the mechanism behind self-consistency [16]. In sample-then-average, every output contributes equally – this is how ensembles [7], Monte Carlo Dropout [11], and test-time augmentation [10] operate. An empirical regularity worth noting: the probability of having at least one correct solution among n samples follows an exponentiated power law [19], yet the practical gains stall earlier because the selection mechanism runs out of discriminative power before generation runs out of variety.

Type III: Tree/Graph Search. Computation branches into a space of partial solutions, with evaluation and pruning at each node. The worst case is $O(b^d)$ for branching factor b and depth d, though pruning makes real costs far lower. Beam search keeps w candidate sequences alive at each decoding step [31]. Monte Carlo tree search repeatedly selects, expands, simulates, and backpropagates – the engine behind the AlphaGo family [6], and more recently applied to LLM reasoning via RAP [23]. Tree of Thoughts lets the language model itself evaluate and branch over reasoning steps [17]. Classical algorithms like A^* fit here too: they spend search effort to drive down solution cost.

Type IV: Test-Time Model Adaptation. Rather than generating additional outputs, these methods spend computation on updating the model's own parameters before or during prediction. Test-time training runs a self-supervised auxiliary loss on each test sample [28]. TENT minimizes the model's output entropy by adjusting batch normalization statistics at test time [29]. MAML is designed from the start in such a way that a few gradient steps on a small support set will specialize the model to a new task [30]. In-context learning occupies a borderline position: appending more demonstrations

to the prompt raises the inference cost and lifts performance, yet no parameters actually change [19]. From a compute-accounting perspective it fits the definition either way.

Type V: Guided Generation. An external signal steers the generation process step by step, adding computational overhead at every iteration. In diffusion, classifier guidance feeds in gradients from a separately trained classifier [25]; classifier-free guidance gets the same effect by running both a conditional and an unconditional forward pass per step [26]. For language models, process reward models score partial reasoning traces to direct search [18]. Retrieval-augmented approaches pull in relevant documents at inference time, expanding the input at additional cost. What limits all of these is the quality of the guiding signal itself – a mediocre classifier or a noisy reward model caps the benefit regardless of how many steps are spent.

Type VI: Adaptive Computation Routing. In this type the total budget does not grow; it is reallocated. Easy inputs get less computation, hard ones get more. Early-exit architectures let confident predictions skip deeper layers. Mixture-of-experts models route different tokens through different parameter subsets. Cascades send queries through progressively larger models, stopping at the first one that is confident enough. Speculative decoding drafts tokens with a small model and verifies them in batch with the full one, cutting average cost without changing the output distribution. Budget-controlled reasoning, as in the s1 approach, enforces a minimum thinking allocation per query [22]. What distinguishes Type VI from the rest is that the compute is not increased – it is spent more deliberately.

Cross-domain observations. The taxonomy, taken as a whole, reveals four regularities that cut across types and domains.

The cost-quality curve is sublinear everywhere. Add denoising steps to a diffusion model, reasoning tokens to a language model, trees to a random forest, or simulations to MCTS – quality grows logarithmically or as a power law with an exponent below one. The consistency of this pattern across unrelated architectures suggests it is an intrinsic feature of iterative refinement rather than a characteristic of any single system.

All selection-based methods – the sample-then-select variant of Type II and all of Type III – hit a verifier bottleneck. The pool of candidate outputs keeps expanding with additional samples, but the fraction of that pool that the verifier can correctly distinguish plateaus [19]. Where verification is automated – unit tests for code, formal proof checkers, game outcomes with known rules – the bottleneck is milder. Where it relies on learned models, verification becomes the constraint that binds first.

A sequential–parallel duality runs through every domain considered. Chain-of-thought and self-consistency are the sequential and parallel faces of the same coin in language models. Denoising steps and noise-seed search play analogous roles in diffusion. Depth and breadth compete for the same budget in any tree search. Evidence from language models shows that easy instances favor sequential refinement while hard ones benefit from parallel exploration [15]. The structural prerequisites for this trade-off – diminishing marginal returns on depth, independent variance across parallel draws – are present wherever inference-time scaling operates, though empirical confirmation outside NLP remains to be done.

Finally, methods travel between domains once their inference-time scaling character is recognized. MCTS originated in board games and migrated to LLM reasoning [23]. The majority-vote logic of self-consistency resurfaced in noise-level search for diffusion [27]. At the level of computational structure, classifier guidance in diffusion and reward-model-steered decoding in language models are the same thing. The classification developed here provides a systematic way to identify these transfers and, more importantly, to flag directions in which transfer has not yet been tried.

Conclusions

This study put forward a formal definition and a six-type classification of inference-time scaling methods that work across the whole of machine learning. The taxonomy, organized by computational topology, covers sequential refinement, parallel sampling with aggregation, tree/graph search, test-time model adaptation, guided generation, and adaptive computation routing. Methods ranging from classical ensembles and beam search to Monte Carlo Tree Search, test-time training, chain-of-thought reasoning, and diffusion denoising were placed within this framework through a three-part decomposition into generation policy, selection policy, and allocation policy.

Four regularities surfaced in the analysis: a sublinear cost-quality curve observed across all six types, a verifier bottleneck capping the gains from selection-based strategies, a sequential–parallel duality running through every domain examined, and documented instances where methods moved from one domain to another – a process the taxonomy makes easier to pursue deliberately. The classification also identified a set of well-known methods – test-time augmentation, Monte Carlo Dropout, ensemble averaging, in-context learning – that fall under the proposed definition of inference-time scaling despite not being conventionally treated that way.

The work grows out of our earlier analysis of inference-time scaling in generative models [13, 14] and extends it to machine learning as a whole. Open questions for future research include empirical testing of the sequential–parallel duality in domains beyond NLP, theoretical foundations for the sublinear scaling law, composite strategies that mix multiple topology types in one system, and models that learn to pick their own inference-time regime on a per-input basis.

Bibliography

1. Kaplan J., McCandlish S., Henighan T. J., Brown T. B., Chess B., Child R., Gray S., Radford A., Wu J., Amodei D. Scaling Laws for Neural Language Models. ArXiv. 2020. arXiv:2001.08361. DOI: https://doi.org/10.48550/arXiv.2001.08361

2. Hoffmann J., Borgeaud S., Mensch A., Buchatskaya E., Cai T., Rutherford E.,... Sifre L. An empirical analysis of compute-optimal large language model training. Advances in neural information processing systems. 2022. Vol. 35, pp. 30016-30030. DOI: https://dl.acm.org/doi/10.5555/3600270.3602446

3. Wei J., Wang X., Schuurmans D., Bosma M., Xia F., Chi E.,... Zhou D. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems. 2022. Vol. 35, pp. 24824-24837. DOI: https://doi.org/10.48550/arXiv.2201.11903

4. Ho J., Jain A., Abbeel P. Denoising diffusion probabilistic models. Advances in neural information processing systems. 2022. Vol. 33, pp. 6840-6851. DOI: https://doi.org/10.48550/arXiv.2006.11239

5. Lipman Y., Chen R. T., Ben-Hamu H., Nickel M., Le M. Flow matching for generative modeling. arXiv. 2022. arXiv:2210.02747. DOI: https://doi.org/10.48550/arXiv.2210.02747

6. Silver D., Huang A., Maddison C. J., Guez A., Sifre L., Van Den Driessche G.,... Hassabis D. Mastering the game of Go with deep neural networks and tree search. Nature. 2026. Vol. 529(7587), pp. 484-489. DOI: https://doi.org/10.1038/nature16961

7. Breiman L. Random forests. Machine learning. 2001. Vol. 45(1), pp. 5-32. DOI: https://doi.org/10.1023/A:1010933404324

8. Welleck S., Bertsch A., Finlayson M., Schoelkopf H., Xie A., Neubig G., Kulikov I., Harchaoui Z. From Decoding to Meta-Generation: Inference-time Algorithms for Large Language Models. ArXiv. 2024. abs/2406.16838. DOI: https://doi.org/10.48550/arXiv.2406.16838

9. Balachandran V., Chen J., Chen L., Garg S., Joshi N., Lara Y.,... Yousefi S. Inference-time scaling for complex tasks: Where we stand and what lies ahead. arXiv. 2025. arXiv:2504.00294. DOI: https://doi.org/10.48550/arXiv.2504.00294

10. Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012. Vol. 25. DOI: https://doi.org/10.1145/3065386

11. Gal Y., Ghahramani Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning. PMLR. 2016. pp. 1050-1059. DOI: https://dl.acm.org/doi/10.5555/3045390.3045502

12. Brown T., Mann B., Ryder N., Subbiah M., Kaplan J. D., Dhariwal P.,... Amodei D. Language models are few-shot learners. Advances in neural information processing systems. 2020. Vol. 33, pp. 1877-1901. DOI: https://doi.org/10.48550/arXiv.2005.14165

13. Бондар В. В., Бабенко В. Г. Масштабування обчислень під час генерації як універсальний принцип для генеративних моделей. Телекомунікаційні та інформаційні технології. 2025. № 4. С. 229–234. DOI: https://doi.org/10.31673/2412-4338.2025.048926

14. Bondar V., Babenko V., Trembovetskyi R., Korobeinyk Y., Dzyuba V. Deep generative models as the probability transformation functions. arXiv. 2025. arXiv:2506.17171. DOI: https://doi.org/10.48550/arXiv.2506.17171

15. Snell C.V., Lee J., Xu K., Kumar A. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. ArXiv. 2024. abs/2408.03314. DOI: https://doi.org/10.48550/arXiv.2408.03314

16. Wang X., Wei J., Schuurmans D., Le Q., Chi E., Narang S.,... Zhou D. Self-consistency improves chain of thought reasoning in language models. arXiv. 2022. arXiv:2203.11171. DOI: https://doi.org/10.48550/arXiv.2203.11171

17. Yao S., Yu D., Zhao J., Shafran I., Griffiths T., Cao Y., Narasimhan K. Tree of thoughts: Deliberate problem solving with large language models. Advances in neural information processing systems. 2023. Vol. 36, pp. 11809–11822. DOI: https://dl.acm.org/doi/10.5555/3666122.3666639

18. Lightman H., Kosaraju V., Burda Y., Edwards H., Baker B., Lee T.,... Cobbe K. Let's verify step by step. In The Twelfth International Conference on Learning Representations. 2023. DOI: https://doi.org/10.48550/arXiv.2305.20050

19. Brown B., Juravsky J., Ehrlich R., Clark R., Le Q. V., Ré C., Mirhoseini A. Large language monkeys: Scaling inference compute with repeated sampling. arXiv. 2024. arXiv:2407.21787. DOI: https://doi.org/10.48550/arXiv.2407.21787

20. Jaech A., Kalai A., Lerer A., Richardson A., El-Kishky A., Low A.,... Metz L. Openai o1 system card. arXiv. 2024. arXiv:2412.16720. DOI: https://doi.org/10.48550/arXiv.2412.16720

21. Guo D., Yang D., Zhang H., Song J., Zhang R., Xu R.,... He Y. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. Nature. 2025. Vol. 645, pp. 633–638. DOI: https://doi.org/10.1038/s41586-025-09422-z

22. Muennighoff N., Yang Z., Shi W., Li X. L., Fei-Fei L., Hajishirzi H.,... Hashimoto T. B. s1: Simple test-time scaling. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. 2025. pp. 20286–20332. DOI: https://doi.org/10.18653/v1/2025.emnlp-main.1025

23. Hao S., Gu Y., Ma H., Hong J., Wang Z., Wang D., Hu Z. Reasoning with language model is planning with world model. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. 2023. pp. 8154–8173. DOI: https://doi.org/10.18653/v1/2023.emnlp-main.507

24. Song Y., Sohl-Dickstein J., Kingma D. P., Kumar A., Ermon S., Poole B. Score-Based Generative Modeling through Stochastic Differential Equations. In International Conference on Learning Representations. 2020. DOI: https://doi.org/10.48550/arXiv.2011.13456

25. Dhariwal P., Nichol A. Diffusion models beat gans on image synthesis. Advances in neural information processing systems. 2021. Vol. 34, pp. 8780-8794. DOI: https://doi.org/10.48550/arXiv.2105.05233

26. Ho J., Salimans T. Classifier-free diffusion guidance. arXiv. 2022. DOI: arXiv:2207.12598. https://doi.org/10.48550/arXiv.2207.12598

27. Ma N., Tong S., Jia H., Hu H., Su Y. C., Zhang M.,... Xie S. Inference-time scaling for diffusion models beyond scaling denoising steps. arXiv. 2025. DOI: arXiv:2501.09732. https://doi.org/10.48550/arXiv.2501.09732

28. Sun Y., Wang X., Liu Z., Miller J., Efros A., Hardt M. Test-time training with self-supervision for generalization under distribution shifts. In International conference on machine learning. 2020. PMLR. pp. 9229-9248. DOI: https://doi.org/10.48550/arXiv.1909.13231

29. Wang D., Shelhamer E., Liu S., Olshausen B. A., Darrell T. Tent: Fully Test-Time Adaptation by Entropy Minimization. International Conference on Learning Representations. 2021. DOI: https://doi.org/10.48550/arXiv.2006.10726

30. Finn C., Abbeel P., Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In International conference on machine learning. 2017. PMLR. pp. 1126-1135. DOI: https://doi.org/10.48550/arXiv.1703.03400

31. Sutskever I., Vinyals O., Le Q. V. Sequence to sequence learning with neural networks. Advances in neural information processing systems. 2014. Vol. 27. DOI: https://doi.org/10.48550/arXiv.1409.3215

32. Jumper J. M., Evans R., Pritzel A., Green T., Figurnov M., Ronneberger O., … Hassabis D. Highly accurate protein structure prediction with AlphaFold. Nature. 2021. Vol. 596, pp. 583–589. DOI: https://doi.org/10.1038/s41586-021-03819-2

33. Li Y., Choi D., Chung J., Kushman N., Schrittwieser J., Leblond R.,... Vinyals O. Competition-level code generation with alphacode. Science. 2022. Vol. 378(6624), pp. 1092-1097. DOI: https://doi.org/10.1126/science.abq1158

References

1. Kaplan, J., McCandlish, S., Henighan, T.J., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling Laws for Neural Language Models. ArXiv, abs/2001.08361. https://doi.org/10.48550/arXiv.2001.08361

2. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E.,... & Sifre, L. (2022). An empirical analysis of compute-optimal large language model training. Advances in neural information processing systems, 35, 30016-30030. https://dl.acm.org/doi/10.5555/3600270.3602446

3. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E.,... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35, 24824-24837. https://doi.org/10.48550/arXiv.2201.11903

4. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. Advances in neural information processing systems, 33, 6840-6851. https://doi.org/10.48550/arXiv.2006.11239

5. Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., & Le, M. (2022). Flow matching for generative modeling. arXiv preprint arXiv:2210.02747. https://doi.org/10.48550/arXiv.2210.02747

6. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G.,... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484-489. https://doi.org/10.1038/nature16961

7. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

8. Welleck, S., Bertsch, A., Finlayson, M., Schoelkopf, H., Xie, A., Neubig, G., Kulikov, I., & Harchaoui, Z. (2024). From Decoding to Meta-Generation: Inference-time Algorithms for Large Language Models. ArXiv, abs/2406.16838. https://doi.org/10.48550/arXiv.2406.16838

9. Balachandran, V., Chen, J., Chen, L., Garg, S., Joshi, N., Lara, Y.,... & Yousefi, S. (2025). Inference-time scaling for complex tasks: Where we stand and what lies ahead. arXiv preprint arXiv:2504.00294. https://doi.org/10.48550/arXiv.2504.00294

10. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25. https://doi.org/10.1145/3065386

11. Gal, Y., & Ghahramani, Z. (2016, June). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning (pp. 1050-1059). PMLR. https://dl.acm.org/doi/10.5555/3045390.3045502

12. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P.,... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877–1901. https://doi.org/10.48550/arXiv.2005.14165

13. Bondar, V. V., & Babenko, V. G. (2025). Inference-time computational scaling calculations as a universal principle for generative models, (4), 229–234. [in Ukrainian] https://doi.org/10.31673/2412-4338.2025.048926

14. Bondar, V., Babenko, V., Trembovetskyi, R., Korobeinyk, Y., & Dzyuba, V. (2025). Deep generative models as the probability transformation functions. arXiv preprint arXiv:2506.17171. https://doi.org/10.48550/arXiv.2506.17171

15. Snell, C.V., Lee, J., Xu, K., & Kumar, A. (2024). Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. ArXiv, abs/2408.03314. https://doi.org/10.48550/arXiv.2408.03314

16. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S.,... & Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171. https://doi.org/10.48550/arXiv.2203.11171

17. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., & Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. Advances in neural information processing systems, 36, 11809–11822. https://dl.acm.org/doi/10.5555/3666122.3666639

18. Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T.,... & Cobbe, K. (2023, May). Let's verify step by step. In The Twelfth International Conference on Learning Representations. https://doi.org/10.48550/arXiv.2305.20050

19. Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Ré, C., & Mirhoseini, A. (2024). Large language monkeys: Scaling inference compute with repeated sampling. arXiv preprint arXiv:2407.21787. https://doi.org/10.48550/arXiv.2407.21787

20. Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A.,... & Metz, L. (2024). Openai o1 system card. arXiv preprint arXiv:2412.16720. https://doi.org/10.48550/arXiv.2412.16720

21. Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R.,... & He, Y. (2025). Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. Nature 645, 633–638. https://doi.org/10.1038/s41586-025-09422-z

22. Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H.,... & Hashimoto, T. B. (2025, November). s1: Simple test-time scaling. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (pp. 20286-20332). https://doi.org/10.18653/v1/2025.emnlp-main.1025

23. Hao, S., Gu, Y., Ma, H., Hong, J., Wang, Z., Wang, D., & Hu, Z. (2023, December). Reasoning with language model is planning with world model. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (pp. 8154-8173). https://doi.org/10.18653/v1/2023.emnlp-main.507

24. Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. In International Conference on Learning Representations. https://doi.org/10.48550/arXiv.2011.13456

25. Dhariwal, P., & Nichol, A. (2021). Diffusion models beat gans on image synthesis. Advances in neural information processing systems, 34, 8780-8794. https://doi.org/10.48550/arXiv.2105.05233

26. Ho, J., & Salimans, T. (2022). Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598. https://doi.org/10.48550/arXiv.2207.12598

27. Ma, N., Tong, S., Jia, H., Hu, H., Su, Y. C., Zhang, M.,... & Xie, S. (2025). Inference-time scaling for diffusion models beyond scaling denoising steps. arXiv preprint arXiv:2501.09732. https://doi.org/10.48550/arXiv.2501.09732

28. Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., & Hardt, M. (2020, November). Test-time training with self-supervision for generalization under distribution shifts. In International conference on machine learning (pp. 9229–9248). PMLR. https://doi.org/10.48550/arXiv.1909.13231

29. Wang, D., Shelhamer, E., Liu, S., Olshausen, B. A., & Darrell, T. (2021). Tent: Fully Test-Time Adaptation by Entropy Minimization. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.2006.10726

30. Finn, C., Abbeel, P., & Levine, S. (2017, July). Model-agnostic meta-learning for fast adaptation of deep networks. In International conference on machine learning (pp. 1126–1135). PMLR. https://doi.org/10.48550/arXiv.1703.03400

31. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27. https://doi.org/10.48550/arXiv.1409.3215

32. Jumper, J. M., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., … & Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. Nature, 596, 583–589. https://doi.org/10.1038/s41586-021-03819-2

33. Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R.,... & Vinyals, O. (2022). Competition-level code generation with alphacode. Science, 378(6624), 1092-1097. https://doi.org/10.1126/science.abq1158

Дата першого надходження статті до видання: 12.01.2026
Дата прийняття статті до друку після рецензування: 16.02.2026
Дата публікації (оприлюднення) статті: 30.04.2026