14 Reasoning Training & Inference-Time Scaling

At the 2016 edition of the Neural Information Processing Systems (NeurIPS) conference, Yann LeCun first introduced his now-famous cake metaphor for where learning happens in modern machine learning systems:

> If intelligence is a cake, the bulk of the cake is unsupervised learning, the icing on the cake is supervised learning, and the cherry on the cake is reinforcement learning (RL).

This analogy is now largely complete with modern language models and recent changes to the post-training stack. In this analogy:

- Self-supervised learning on vast swaths of internet data makes up the majority of the cake (especially when viewed in compute spent in FLOPs),
- The beginning of post-training in supervised finetuning (SFT) for instructions tunes the model to a narrower distribution (along with the help of chosen examples for RLHF), and
- Finally "pure" reinforcement learning (RL) is the cherry on top.

We learn just "a few bits" of information with RL in just a few training samples. This little bit of reasoning training emerged with **reasoning models** that use a combination of the post-training techniques discussed in this book to align preferences along with RL training on verifiable domains to dramatically increase capabilities such as reasoning, coding, and mathematics problem solving.

The training method for these models, Reinforcement Learning with Verifiable Rewards (RLVR) [6], proceeds very similarly to RLHF, but it makes the reward model optional in lieu of a scoring function that returns a positive reward when the answer is correct and 0 otherwise.

The first models to successfully deploy this type of training were OpenAI's o1 [47] and the open-weight model DeepSeek R1 [138]. Soon after, the entire AI industry prioritized this training process and model style. The core change here is more of a reallocation of the stages of training and the priority of different behaviors rather than this type of RL setup being entirely new. Reasoning models brought an era where scaling RL training is expected.

As for the type of behavior these models accrue, consider the following example with DeepSeek V3 0325 versus their reasoning model, DeepSeek R1, on the query `Write me a short poem about a goldfish`. DeepSeek V3 is very to the point:

> Goldfish Dreams
>
> In a bowl of glass so clear and round, A tiny goldfish swims without a sound. With scales that gleam like sunlight's kiss, He drifts along in watery bliss.
>
> A flick of fin, a flash so bright, Chasing bubbles in the light. Does he dream of oceans wide? Or is his little world enough—his pride?

Meanwhile, DeepSeek R1's experience is different. It first generates numerous thinking tokens to craft a response before responding to the user. The answer appears as:

> `<thinking>` Okay, the user wants a short poem about a goldfish. Let me start by brainstorming some goldfish-related imagery. Goldfish are often in bowls, right?

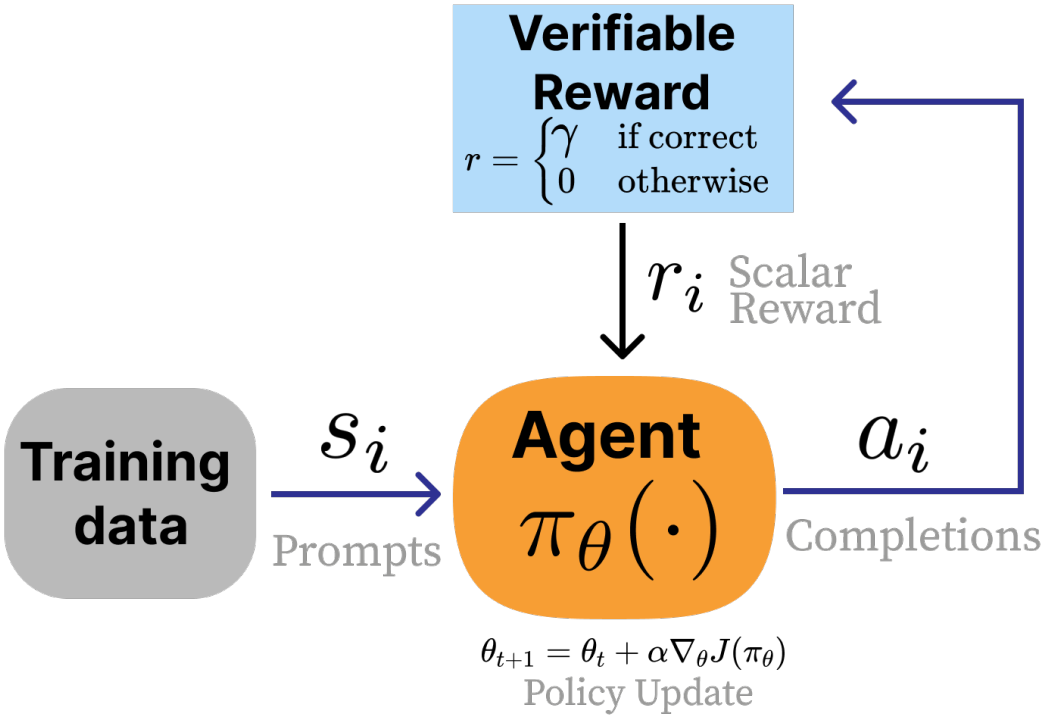

Figure 15: RLVR in the form of an RL feedback loop. Instead of a reward model, we use a verification function.

> So maybe start with the setting. Words like "gilded bowl" or "crystal globe" could work.
>
> Next, think about the movement of the goldfish. They swim smoothly, so words like "dart", "glide", or "swirl" come to mind. Maybe describe their fins as "silken" to emphasize delicacy.
>
> Then, consider the environment around the goldfish. Sunlight through water creates shimmering effects. Words like "sunbeams dance" or "ripples spin" might capture that. Including elements like castles or pebbles adds a playful touch.
>
> Themes? Maybe the contrast between the goldfish's small world and its perception of freedom. The bowl as a universe, tiny but complete. Ending with a sense of mystery or silent observation could give depth. Need to keep it concise, maybe four stanzas with a rhyme scheme. Let me check the rhythm and flow. Avoid clichés but keep it accessible. Alright, let's piece it together. `</thinking>`
>
> Gilded Dreams in Liquid Glass
>
> A gilded bowl, a crystal globe, Where tiny tides in silence flow— A flash of orange, a whispered robe, You dart through sunlit indigo.
>
> Your world is small, yet boundless too— Each pebble, castle, weed unfurled. The surface bends the sky to blue, And ripples spin your liquid world.
>
> You glide through myths we'll never know, A silent monarch, veiled in mist. Do dreams drift where the bubbles go? Or memories in amethyst?
>
> We watch you trace your endless sphere— A universe within a tear.

The tokens in the `<thinking>` tags is the model's reasoning. For more complex problems the reasoning stage can take thousands of tokens before producing an answer. So, long-context language models were a prerequisite to advanced reasoning behavior, but that is not the focus of this chapter.

The core intuition for *how this training works* is that for a given model, we repeat the following cycle:

1. Sample multiple answers to multiple questions,
2. Take gradient steps towards the answers that are correct, and
3. Repeat, revisiting the same data.

Remarkably, this extremely simple approach (when done with a careful distribution of data and stable training infrastructure) helps the models learn by revisiting the same questions again and again. Even more remarkable is that the improvements on these training questions generalize to questions and (some) domains the models have never seen!

This simple approach allows the models to lightly search over behavior space and the RL algorithm increases the likelihood of behaviors that are correlated with correct answers.

14.1 Why Does RL Work Now?

Despite many, many takes that "RL doesn't work yet" [188] or paper's detailing deep reproducibility issues with RL [189], the field overcame it to find high-impact applications.

The takeoff of RL-focused training on language models indicates steps in many fundamental issues for the research area, including:

- **Stability of RL can be solved**: For its entire existence, the limiting factor on RL's adoption has been stability. This manifests in two ways. First, the learning itself can be fickle and not always work. Second, the training itself is known to be more brittle than standard language model training and more prone to loss spikes, crashes, etc. Countless releases are using this style of RL training and substantial academic uptake has occurred. The technical barriers to entry on RL are at an all time low.
- **Open-source versions already "exist"**: Many tools already exist for training language models with RLVR and related techniques. Examples include TRL [41], Open Instruct [6], veRL [190], and OpenRLHF [191], where many of these are building on optimizations from earlier in the arc of RLHF and post-training. The accessibility of tooling is enabling a large uptake of research that'll likely soon render this chapter out of date.

Multiple resources point to RL training for reasoning only being viable on leading models coming out from about 2024 onwards, indicating that a certain level of underlying capability was needed in the models before reasoning training was possible.

14.2 RL Training vs. Inference Time Scaling

Training with Reinforcement Learning to elicit reasoning behaviors and performance on verifiable domains is closely linked to the ideas of inference time scaling. Inference-time scaling, also called test-time scaling, is the general class of methods that use more computational power at inference in order to perform better at a downstream tasks. Methods for inference-time scaling were studied before the release of DeepSeek R1 and OpenAI's o1, which both massively popularized investment in RL training specifically. Examples include value-guided sampling [192] or repeated random sampling with answer extraction [193]. Beyond this, inference-time scaling can be used to improve more methods of AI training beyond chain of thought reasoning to solve problems, such as with reward models that consider the options deeply [92] [194].

RL training is a short path to inference time scaling laws being used, but in the long-term we will have more methods for eliciting the inference-time tradeoffs we need for best performance. Training models heavily with RL changes them so that they generate more tokens per response in a way that is strongly correlated with downstream performance. This is a substantial shift from the length-bias seen in early RLHF systems [9], where the human preference training had a side effect of increasing response rate for marginal gains on preference rankings.

Downstream of the RL trained models there are many methods being explored to continue to push the limits of reasoning and inference-time compute. These are largely out of the scope of this book due to their rapidly evolving nature, but they include distilling reasoning behavior from a larger RL trained model to a smaller model via instruction tuning [195], composing more inference calls [196], and more. What is important here is the correlation between downstream performance and an increase in the number of tokens generated – otherwise it is just wasted energy.

14.3 The Future (Beyond Reasoning) of Reinforcement Finetuning

In many domains, these new flavors of RLVR and reinforcement finetuning are much more aligned with the goals of developers by being focused on performance rather than behavior. Standard finetuning APIs generally use a parameter-efficient finetuning method such as LoRA with supervised finetuning on instructions. Developers pass in prompts and completions and the model is tuned to match that by updating model parameters to match the completions, which increases the prevalence of features from your data in the models generations.

Reinforcement finetuning is focused on matching answers. Given queries and correct answers, RFT helps the model learn to get the correct answers. While standard instruction tuning is done with 1 or 2 epochs of loss updates over the data, reinforcement finetuning gets its name by doing hundreds or thousands of epochs over the same few data points to give the model time to learn new behaviors. This can be viewed as reinforcing positive behaviors that would work sparingly in the base model version into robust behaviors after RFT.

The scope of RL training for language models continues to grow: The biggest takeaway from o1 and R1 on a fundamental scientific level was that we have even more ways to train language models to potentially valuable behaviors. The more open doors that are available to researchers and engineers, the more optimism we should have about AI's general trajectory.