Reading Assignment 13: RL

When I think about these questions, a few things come to mind immediately: what kind of behaviors do I actually want the model to learn before I start "optimizing" them with RL, and how fragile is my notion of "goodness" once I encode it into a reward model? How much of what I call "alignment" is really just the side effect of dataset curation plus SFT, and how much is genuinely coming from RL over a learned reward? With that in mind, here is how I see both questions.

For the first question, we include SFT in the RLHF pipeline because RL is a very brittle way to search over behavior space if we start from a raw pretrained model. Pretraining gives a broad, somewhat chaotic landscape of behaviors: useful reasoning patterns are mixed with toxic, off-distribution, and goal-misaligned tendencies. With SFT, we force a sharp projection of that landscape onto a narrow manifold of "plausibly good" responses, defined by the curated instruction-following dataset. That does two things. First, it initializes the policy in a region of behavior space where my reward model will have meaningful signal: the SFT model already looks like a decent assistant, so the difference between "good" and "great" responses is fine-grained and learnable for the reward model. Second, it makes RL optimization much more stable: instead of pushing from a random point, I run policy gradient updates in a small neighborhood around a strong baseline that already aligns with user instructions and basic safety constraints.

If I tried to start with RL directly on top of the pretrained model, I would be asking the reward model to distinguish between wildly variable outputs, many of which are far from anything I would ever want the system to say. That amplifies noise and spurious correlations: the reward model will latch onto superficial patterns that correlate with preference labels in a small region, then generalize badly outside it. Empirically, RL on top of SFT tends to give smoother learning curves, better sample efficiency, and less collapse into degenerate behaviors compared with RL from a purely pretrained checkpoint. The SFT stage also makes exploration safer: the model is already "house-trained," so even when I add entropy for exploration, most outputs remain within a reasonable band of politeness, coherence, and harmlessness. So in practice, SFT is not just a warmup; it is a way of defining the feasible set of behaviors on which RLHF is allowed to operate.

For the second question, I see the main pitfalls of reward modeling as mismatches between what we intend to capture and what the reward actually incentivizes. Reward models are trained on finite, noisy preference data, often with annotator disagreement, positional bias, and context effects. If my dataset over-represents a certain style of answer, tone, or length, the reward model can implicitly treat those superficial features as proxies for "goodness." This leads to known failure modes like reward hacking and output

homogenization: the policy learns to maximize whatever cheap heuristics the reward model finds easy to score, not the deeper notion of helpfulness, honesty, and harmlessness we care about. Another pitfall is label leakage: if annotators implicitly reward "confident, authoritative" tone, the reward model can push the policy toward overconfident hallucinations, because those look good to the scorer, even though they're epistemically misaligned.

Downstream, these issues translate into alignment pathologies. If the reward model systematically undervalues uncertainty, nuance, or minority perspectives, then the aligned model will gradually erase those behaviors as it optimizes for higher reward. If the reward model is poorly calibrated on edge cases, the RLHF policy may look well-aligned on standard benchmarks but behave badly in rare or adversarial settings. Since I use the reward model as an implicit definition of "what we want," any blind spot in that scalar reward becomes a blind spot in the final system's behavior. Over multiple rounds of RL and iteration, we can even get alignment drift: the policy adapts to exploit quirks of the reward model, the distribution of outputs shifts, and the reward model is then retrained on a biased slice of behavior, locking in and amplifying its own earlier mistakes.

So to me, SFT is the structural constraint that keeps RLHF grounded, and the reward model is the approximate lens through which we see "good behavior." If we skip SFT, the lens is pointed at a noisy, unconstrained space and we get chaotic updates. If we build the lens carelessly, we get a model that faithfully optimizes a value function we never truly meant to write down. In both cases, what we call "alignment" becomes fragile, because the system is now tuned to maximize a scalar surrogate and any mismatch between that surrogate and our actual preferences shows up as a systematic behavioral bias.