Predicting vs. Acting: A Trade-off Between World Modeling & Agent Modeling

***Margaret Li**[1,2] ***Weijia Shi**[1,2] **Artidoro Pagnoni**[1,2]
Peter West[1,3] **Ari Holtzman**[2,4]
[1]University of Washington [2]Meta
[3]University of British Columbia
[4]University of Chicago
{margsli,swj0419,artidoro}@cs.washington.edu
pwest@cs.ubc.ca aholtzman@uchicago.edu

Abstract

RLHF-aligned LMs have shown unprecedented ability on both benchmarks and long-form text generation, yet they struggle with one foundational task: next-token prediction. As RLHF models become agent models aimed at interacting with humans, they seem to lose their *world modeling*—the ability to predict what comes next in *arbitrary* documents, which is the foundational training objective of the Base LMs that RLHF adapts.

Besides empirically demonstrating this trade-off, we propose a potential explanation: to perform coherent long-form generation, RLHF models restrict randomness via implicit *blueprints*. In particular, RLHF models concentrate probability on sets of *anchor spans* that co-occur across multiple generations for the same prompt, serving as textual scaffolding but also limiting a model's ability to generate documents that do not include these spans. We study this trade-off on the most effective current agent models, those aligned with RLHF, while exploring why this may remain a fundamental trade-off between models that *act* and those that *predict*, even as alignment techniques improve.

1 Introduction

Alignment via RLHF (Ivison et al., 2023; Touvron et al., 2023) trains models towards action: completing specific goals and excelling across both short and long-form textual tasks. RLHF works by adapting base LMs that are trained to be *world models*, accurately predicting the distribution of text that might occur after an arbitrary prefix. While RLHF models tend to excel at complex tasks, in this work we find that they partially lose the *world modeling* abilities that allow base LMs to simulate documents from the broader distribution of the internet.

We propose that this trade-off is a natural result of RLHF models concentrating probability to specific spans, which allows these models to blueprint long-form generation (Figure 1) but reduces their ability to model arbitrary text.

Specifically, RLHF models struggle with next-token prediction which directly measures ability to world model, even when they are finetuned to regain these skills (§2). RLHF models seem to concentrate probability on a smaller set of text (§3), which follows past work on distributional collapse (Shumailov et al., 2023). Yet this concentration may have a use: in making generation more self-predictable, and helping to blueprint long-form text generations. For example, we find *anchor spans* (Figure 1) which appear across many samples for the same prompt, and seem to serve as scaffolding for generation (§4).

Is a trade-off between world modeling and agent modeling fundamental? We argue that self-predictability is likely an inevitable aspect of successful agent models, not just a spurious byproduct of RLHF. In order to generate coherent long-form responses (or act towards goals in general) an agent must guarantee that its future actions are largely predictable to its current self. In effect, *agent modeling* may require minimizing long-term uncertainty while *world modeling* requires maintaining the true uncertainty of natural text documents, a fundamental trade-off. We briefly explore this question in §5.

Such a trade-off would suggest that methods for adapting models to useful tasks, such as RLHF, will tend to narrow the breadth of a model. In other words, an agent model that takes actions towards a long-term goal may not be fully representative of the broader distribution of all possible agents and goals. General systems covering both sets of abilities might combine agent and world models rather than relying on agent models to both act and predict.

*Authors contributed equally

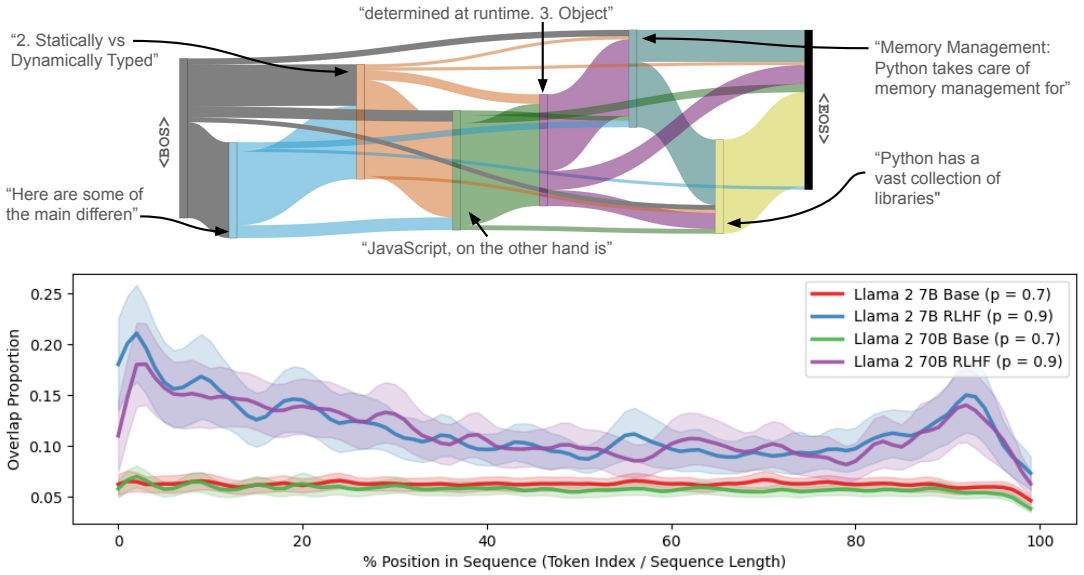


Figure 1: **RLHF model generations on the same prompt are highly similar to each other, unlike Base LMs.** For each of 80 short prompts, we collect and align 100 generations (nucleus sampling, p = 0.9) from Base (pretrained) and RLHF models. **Above:** (§4.2) A Sankey diagram of 100 RLHF model generations for the prompt "What are the main differences between Python and JavaScript programming languages?" Sequences share multiple lengthy anchor spans which appear verbatim in the same order, forming a uniform skeleton for nearly all generations. **Below:** (§4.3) Over the sequence length, the number of generations aligned with at least 5 others, averaged over all prompts. Base model generations maintain low levels of alignment. RLHF model generations exhibit high alignment throughout, but especially near the beginning and end of generations.

2 Agent models aren't general language models anymore

Many current language models begin as *world models*, trained to accurately predict the probabilities of possible events in a medium (i.e. text), and are later adapted as *agent models*–trained to interact with users towards specific goals. While agent models, particularly those trained with RLHF, are unparalleled as interactive dialogue agents, we show in this section that such adaptation diminishes the original ability to world model, i.e., provide accurate estimated probabilities of text. Our analysis shows that agent models significantly underperform the base models they are trained from on a set of language modeling tasks across diverse domains (§2.1). Even when re-trained towards language modeling (§2.2), they fail to match base models, suggesting a potential trade-off between the abilities of *agent* and *world* models.

2.1 Perplexity of Base vs. RLHF models

The ability to accurately predict the what will happen next given arbitrary starting conditions (i.e., to world model) can be evaluated in the text domain as performance on next-token prediction with the perplexity metric. We evaluate the perplexity of the Base models against their agent model counterparts. We focus on RLHF as a means of producing agent models, as this is the most popular and effective approach currently being used. Specifically, we analyze two Base models: Llama 2 (Touvron et al., 2023) and OLMo (Groeneveld et al., 2024). For their RLHF adaptations, we employ Tulu 2 (Ivison et al., 2023) and Llama 2 Chat (Touvron et al., 2023) for Base Llama 2, and OLMo RLHF for the Base OLMo. We consider common pretraining corpora such as C4 (Roberts et al., 2019), Arxiv (Clement et al., 2019), and Wikipedia for evaluation. To test generalization to new data, we also use new corpora released post-model development including new Arxiv papers and BBC news stories (Li et al., 2023b). Furthermore, we incorporate instruction finetuning data like Humpback (Li et al., 2023a) and chat assistant data such as OASST1 (Köpf et al., 2024), Anthropic Harmless and Helpful corpora (Bai et al., 2022).

Results in Figure 2 (A) show that agent models perform consistently worse than the Base models that they were adapted from at language modeling. On standard test sets crawled directly from the broader internet (e.g., C4) or specific domains (e.g., Arxiv) this is not a surprising result: RLHF models

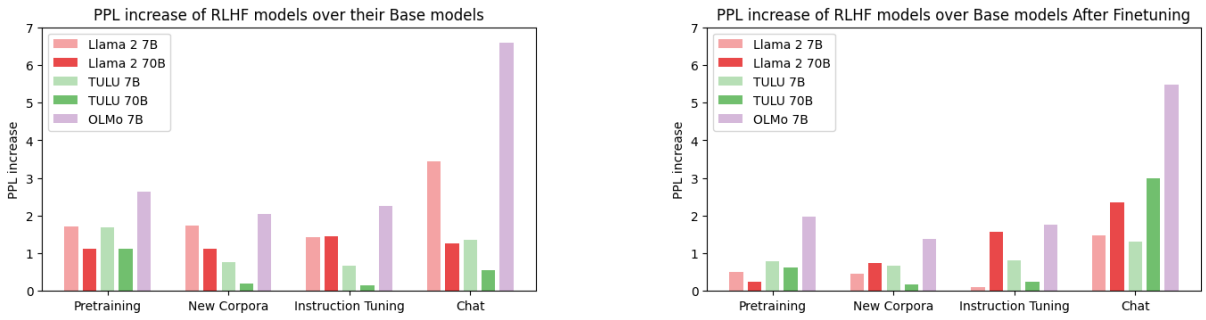


Figure 2: **(§2.1) RLHF models are significantly worse on language modeling tasks compared to the Base LMs they are adapted from, even on data similar to their preference tuning corpora. (A):** Perplexity increase in models post-RLHF compared to the base LLM, across several model families and sizes, evaluated on 9 text corpora grouped into 4 categories. RLHF models consistently underperform the Base models they were tuned from. (The lower the perplexity, the better.) **(B):** We finetune each model from (A) on the target corpus; the general trend remains unchanged. RLHF models are consistently inferior even post-finetuning. Details in Appendix E.

are trained to produce specific distributions, and thus are no longer good density estimators of the language model pretraining data distribution. However, we show that this holds true *even on instruction tuning and chat assistant datasets* which are exclusively aimed at capturing the action-focused data that current RLHF models are trained towards.

RLHF models are poor zero-shot language models, even on distributions they are trained to imitate, suggesting that they are doing something other than simply capturing these distributions. Indeed, this is not the goal of using RLHF, but it raises a question: what are RLHF models actually doing? In §3 we argue that RLHF models shift their distribution towards a space that can reliably produce high-reward responses via *implicit blueprints*. This limits the generality of predictions from language models adapted to be agent models. In other words: adapting models to become agent models reduces their capacity as world models.

2.2 Readaptation via Finetuning

While RLHF certainly warps the distribution of LLMs to be worse predictors, is this merely a surface-level change? Is the next-token-prediction still hidden within the weights of the adapted model? Perhaps the information for arbitrary next token prediction is simply unused in the output layers of RLHF models. Fully addressing this concern is beyond the scope of this paper, but we present evidence that it is at least not *trivial* to recover the next-token-prediction capabilities of RLHF models. To test this hypothesis we continue to pretrain both Base models and RLHF models on the training sets of the evaluation corpora used in §2.1.

The results in Figure 2 (B) show that it is difficult to recover the original ability of RLHF models to act as language models. Note that this remains true *even for instruction tuning and chat assistant corpora*, which much more closely match the distribution that agent models have been adapted to. Further evidence that RLHF models are not even good *rankers* of likely text is shown in Appendix D via the Shannon Game (Hovy and Lin, 1998).

3 Agent models concentrate probability

Shumailov et al. (2023) have suggested that RLHF models collapse their probability distributions, assigning high probability to a small set of tokens rather than a smoother distribution as observed in base language models. In this section we quantify some properties of this collapse, leading us to propose that RLHF concentrates model probability onto text that is *predictable* by the model, yet still diverse and high quality. This will be an important point in §4 when we discuss how agent models appear to have implicit blueprints for generating text by narrowing the scope of possible futures, particularly onto self-predictable text that could make structuring long-form text easier.

Figure 3 shows how concentrated different model distributions are, measured as the probability mass assigned to the most probable k tokens on gold vs. generated data. The most top-heavy distribution is very clearly RLHF on its own sampled generations, e.g., the average probability of the highest probability token in RLHF when conditioning on its own generated text is nearly 0.9

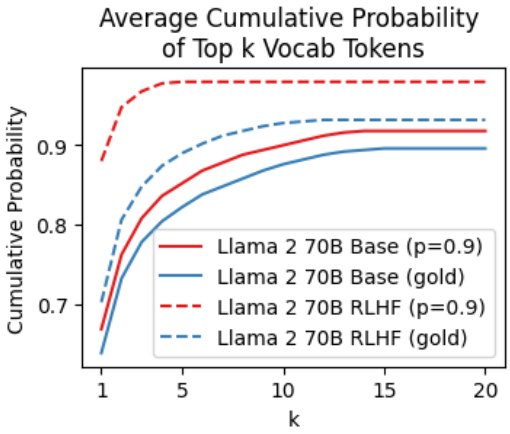


Figure 3: **(§3) RLHF models assign nearly all of the next-token probability mass to a single token, more than Base models.** For Base and RLHF models, we calculate the next-token probability distributions on the gold sequences, as well as on the models' own generations (nucleus sampling, p=0.9). We show the cumulative probability mass of the tokens sorted in descending order of probability. RLHF models assign a larger portion of the probability mass to a small number of tokens, compared to Base models. Details are in Appendix E.3.

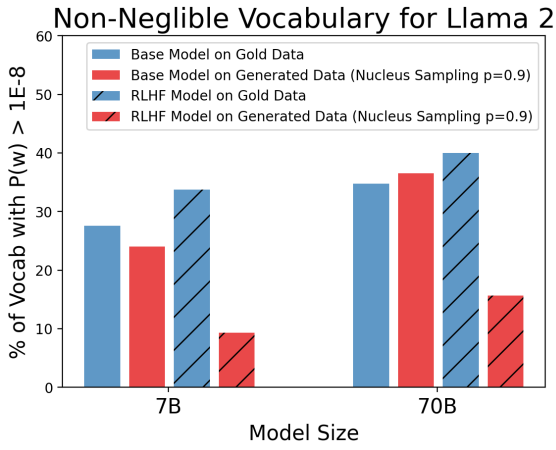


Figure 4: **(§3) RLHF models assign near-zero probability mass to almost all tokens when generating, more than Base models.** For both 7B and 70B models, RLHF models assign non-negligible ($> 10^{-8}$) probability to *significantly fewer* vocabulary tokens when predicting next tokens from its own generations (nucleus sampling, p = 0.9), but slightly *more* tokens when predicting next tokens on gold text sequences. This suggests RLHF models only exhibit collapse in their own generative distributions. Details are in Appendix E.3.

whereas it is well below 0.7 for both Base model setups. Interestingly, it is not only RLHF vs. Base models which exhibits a noticeable gap, but also RLHF models conditioned on gold vs. generated data. When RLHF models are conditioned on their generated text, they become extremely confident, suggesting that RLHF models remain in a region of confidence once generation has started.

We can also consider the converse: how *diffuse* model distributions are, measured by how many tokens are assigned non-negligible probability. Figure 4 displays the average number of vocabulary items with greater than 10^{-8} probability per timestep; in other words, the average number of tokens with a non-trivial probability of being sampled at generation time, which we can also think of as a rough measure of generative uncertainty. As the figure shows, RLHF models have vocabulary distributions that tend to be slightly heavier tailed (more uncertain) on human authored data, but much *lighter tailed* (less uncertain) when conditioned on their own generated text. Interestingly, Base models do not differ as drastically between human authored gold data and their own generations, and RLHF on gold data is similar to Base model uncertainty but slightly *more* uncertain than Base models. The fact that RLHF models have such low uncertainty on their own data, but otherwise very high uncertainty, suggests that they are highly self-conditional and adapted to predicting their own distributions, limiting diversity but allowing for long-term predictability that can aid long-form generation. We explore what this predictability looks like in §4.

This is not merely a result of the low diversity of RLHF models. Adjusting Llama 2 70B RLHF (with nucleus sampling, p=0.9) to be roughly as diverse as Llama 2 70B Base (see $n$gram statistics in Table 4), we still find Llama 2 70B Base is less confident, even on its own *greedy* generations (see Figure 16 in Appendix). In effect, RLHF models seem to be better at self-prediction, even accounting for diversity.

These results suggest that RLHF models may be staying within a region of confidence when they generate text, in-turn allowing for better long-form coherency. It seems that agent models are significantly better at predicting their own generated text than next token predictors, which is verified in Figure 5. This naturally leads to the hypothesis that this kind of high confidence is required to think ahead for long-form generation (§4), implying that the observed tradeoff may be fundamental between *acting* and *predicting*.

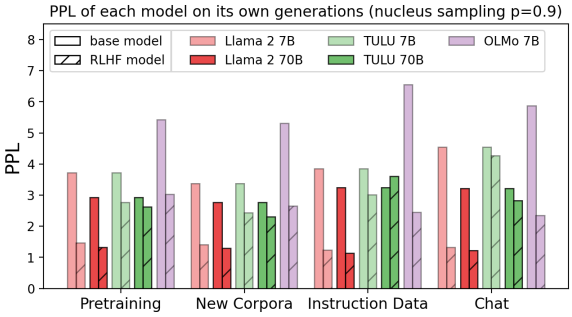


Figure 5: **(§3) RLHF models have lower perplexity when evaluated on their own generations**: We generate (nucleus sampling, p=0.9) completions of prefixes taken from 11 datasets, grouped into 4 categories. Details are in Appendix 3.

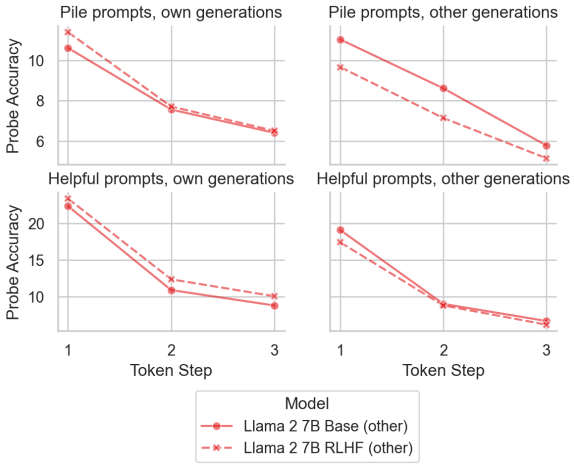


Figure 6: **(§4.1) The intermediate representations of RLHF models contain more linearly extractible information for predicting future tokens of generated text, compared to Base models.** We train a linear probe to predict each model's future tokens ($n = 1, 2, 3$ tokens into the future) given the hidden representation. Each subplot indicates probe accuracy at predicting the model-generated token n steps in the future. **Top** uses prompts from the Pile, and **Bottom** uses prompts from the Anthropic Helpful dataset. **Left** predicts the model's own generation, while **Right** predicts for the other model (e.g. RLHF predicting Base and vice versa).

4 Agent models think ahead

We have shown that RLHF models are no longer good next token predictors and that they concentrate their probability distributions into more predictable outcomes. What does this probability mass shift lead to? In this section we show evidence that RLHF models make use of a kind of implicit blueprint for long-form generation rather than predicting only one token in advance, effectively using probability concentrated on certain future spans to enable "thinking ahead" in long form generation.

4.1 RLHF hidden states are more predictive of future tokens

As a basic measure of "thinking ahead", we estimate the information models have about future timesteps, beyond the next token. Using the methodology of Pal et al. (2023), we study how well we can predict tokens for Llama 2 7B Base and RLHF models by leveraging their own hidden states and a linear probe. Specifically, we perform evaluation on the Pile (Gao et al., 2020) and Anthropic Helpful (Bai et al., 2022) datasets. We train a linear model on the hidden representation from the 28th layer (Llama 2 7B comprises 32 layers), using 100,000 tokens from each dataset, to predict tokens n steps into the future ($n = 1, 2, 3$), *after* the token being predicted at the current timestep. We use this prediction accuracy as a metric to assess the effectiveness of using a model's hidden state to foresee its own generation of future tokens.

Linear probes on RLHF models can predict future tokens in their own generations with higher accuracy than Base models (Figure 6). RLHF adapted models find it easier to predict the future because they generate a future that is more predictable, as supported by Figure 5.

Figure 6 also shows that RLHF models may contain a subset of information that base LLMs do, as base LLMs are more effective at predicting future tokens of RLHF models than the reverse. This supports the idea of concentration or collapse in RLHF models, as their predictions fit base LLMs, but they no longer explain base LLM predictions as effectively.

4.2 Anchor spans as an implicit blueprint

Besides containing more information about next tokens, RLHF models seem to qualitatively *blueprint* generations, consistently using the same key points across many generations for a given prompt (Figure 1 top). Base LLMs show no such structure (Figure 8 bottom).

We visualize these blueprints with Sankey diagrams (Figure 1 top), which characterize the flow through critical nodes over a sequential process. We focus on "anchor spans"—substrings that occur across many sampled outputs for the same prompt—such that the flow between two nodes A and B represents the set of generations which contain A, followed by B, with (possibly different) text in between. Specifically, we first identify all

text spans of a fixed minimum length (30 characters), which occur at a specific index for at least some threshold number of generations (20%). If the same span occurs at two different positions in the alignment indexing, they are counted as different spans. We sort the spans in descending order of frequency across unique outputs, and then by span length. We then greedily pick up to a maximum (6 in these visualizations) number of spans, updating the occurrence counts for unpicked spans after each addition, so that generations cannot "double count" for spans which overlap. See Appendix E.4 for further details.

Qualitatively, RLHF models seem to blueprint their long-form generations: Figure 8 (top) shows that anchor spans remain constant across many outputs, analogous to a bullet point outline embedded within a generation. For the same prompt the Base model (Llama 2 70B) has no such structure whatsoever, even when using a more restricted sampling strategy (Figure 8 bottom). In other words, RLHF models rely on an implicit blueprint, converging to certain spans from which they can reliably predict their own future, while Base models diverge in unpredictable ways (more examples Appendix A).

4.3 Sampled generations contain align-able backbones

The blueprints in §4.2 appear quantitatively as well, as the degree of alignment possible across multiple generations. Sampling 100 continuations for the same prompt from an RLHF model leads to sequences with a significant amount of diversity (in terms of unique $n$grams), but a surprisingly high amount of overlap between the sequences. While diversity and overlap may seem to be in conflict, Figure 7 shows this is not the case. We can use nucleus sampling (Holtzman et al., 2020) to generate text with similar $n$gram diversity across Base LLMs and RLHF-adapted models by setting the value of p appropriately (see Appendix E.4). Yet even when diversity statistics are similar, RLHF models reuse more long $n$grams across different generations for the same prompt than models not trained using Reinforcement Learning. In effect, the diversity is not evenly distributed in RLHF generation, with diversity concentrated in between predictable, long spans.

In Figure 7, on average a quarter of RLHF outputs share at least one 25-gram (about a sentence), and there are almost twice as many RLHF sequences sharing at least a 10-gram than for the Base model. This high level of *n*gram sharing is not limited to a few isolated cases or to the prefix or suffix of the generations. The bar-chart on the right side of Figure 7 (B) shows that on average 10-grams are shared by more sequences in RLHF model outputs by a large margin (note log-scale).

However, *n*gram statistics are not robust to small variations, nor do they consider ordering effects—where *n*grams in RLHF models tend to occur in certain orders. To handle these problems, we propose to first align the continuations for a given prompt sequence alignment software originally developed for bioinformatics, MAFFT (Katoh and Standley, 2013).

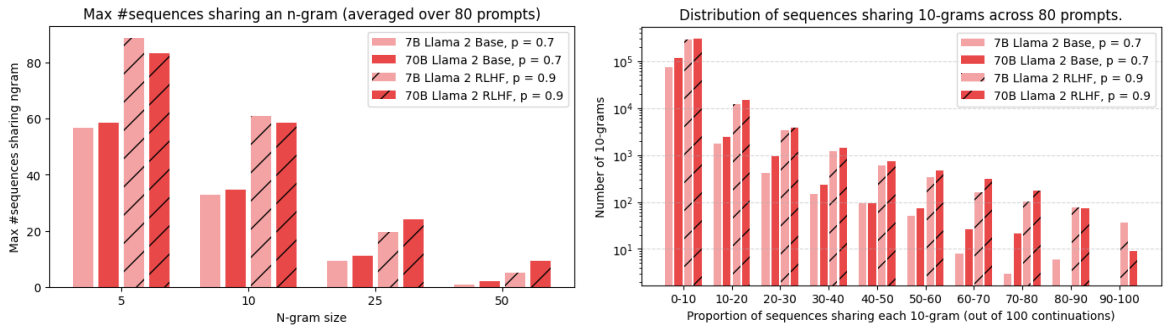
After aligning 100 continuations for each prompt (Figure 1, bottom), Base models have a much lower overlap compared to RLHF models despite having controlled for *n*gram diversity using the nucleus sampling parameter p, as described in Appendix E.4. RLHF models have significantly more overlap for all positions, and at the beginning and end of the sequences the overlap is even more drastic. This shows that RLHF models tend to converge back towards the end of the generations. Additionally, we note that the larger confidence intervals (shaded in the figure) indicate that the convergence points occur at different positions in the sequences for different prompts. These results hold true across different models, see Appendix B.

5 Is this a fundamental trade-off?

Are these differences in world models and agent models fundamental, or just a result of current language model training and RLHF practices? Even if new agent adaptation methods alleviate some of these problems, we propose that, under fixed capacity, there must necessarily be some trade-off between world models and agent models because of the underlying differences in their optimal behavior, which we are more closely approaching with recent state-of-the-art methods.

5.1 Why can't world models just be agent models when we ask them to be?

Prompts almost never fully specify desired behavior. Base LLMs have been trained to sample from the space of *possible documents with a given prefix*. In the overwhelming majority of cases, prefixes do not fully specify many of the choices a document can represent. Furthermore, Base LLMs have been shown to be very sensitive to short-term



(A) **(B)**

Figure 7: **(§4.3) Given a short prompt, RLHF models heavily reuse n-grams across many independently generated continuations (nucleus sampling, p = 0.9), with the most common 10-grams appearing in 60% of generations on average.** For each of 80 short prompts, we collect 100 generations from Base and RLHF models using nucleus sampling with p = 0.7 and p = 0.9, respectively. **(A)** For each prompt, the number of generations, out of 100 total, which contain the most common n-gram ($n \in [5, 10, 25, 50]$), averaged across all prompts. **(B)** A histogram, binning 10-grams by the number of sequences containing that 10-gram. Counts are log-scale. Compared to Base models, RLHF models much more frequently generate the same 10-gram in nearly all continuations for a prompt. Statistics for other models are available in Appendix C.

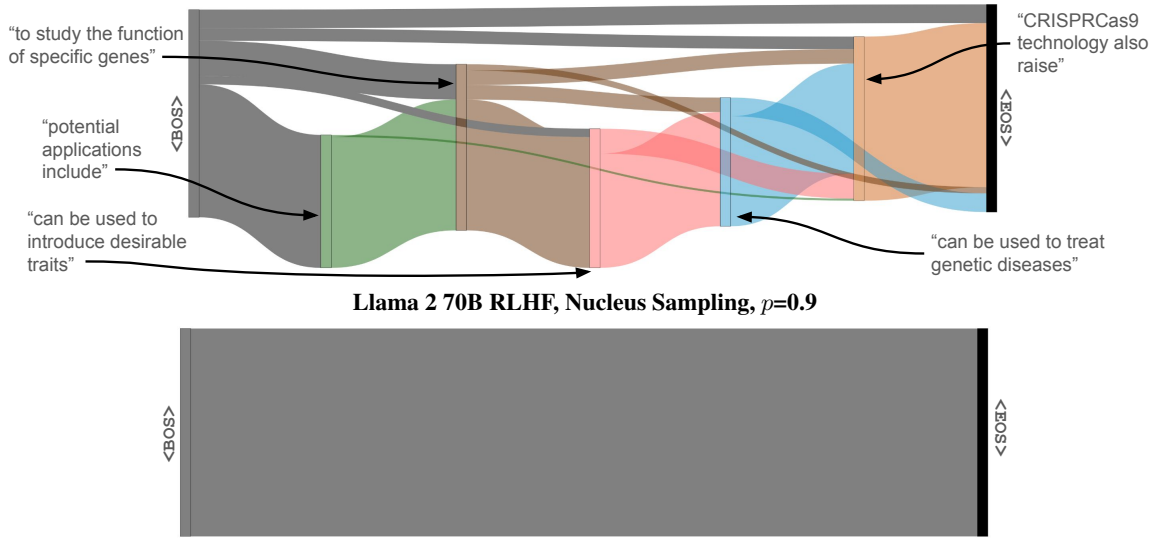


Figure 8: **(§4.2) The RLHF model (top) shows multiple anchor spans appearing verbatim in the same order, whereas the Base model (bottom) lacks any common anchored points, highlighting the divergence of its generated responses.** Each Sankey diagram visualizes 100 responses (nucleus sampling, $p = 0.9$ for RLHF and $p = 0.7$ for Base) to the prompt "Explain the process of gene editing using CRISPR-Cas9 technology, and discuss its potential applications and ethical implications."

context (Paperno et al., 2016; Wang et al., 2023). This is a feature, not a bug, of world models, as short-term context is generally more predictive than long-term context in human-authored documents. Yet it means that Base LLMs are highly sensitive to sampling even one incoherent token. Dziri et al. (2024) suggest that the probability of generating an error is lower bounded by the $1 - P(\text{error})^{\ell}$ where ℓ is the length of a document. The actual error rate is likely much higher, as the model conditions on previously made errors, creating a snowball effect, as with hallucination (Zhang et al., 2023).

5.2 Planning around randomness

The ability for agents to plan well is directly related to the amount of randomness in their environment. Reinforcement Learning (RL) is the mathematical language of agent modeling, so we take a moment to conceptualize why RLHF models would collapse and rely on anchor spans in terms of RL. Recall the

Bellman equation, describing the value of a current state of a sequence of actions:

$$V(s) = \max_{a \in A} \left\{ R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V(s') \right\} \tag{1}$$

where V is the value of state s, A is the set of possible actions, R is the reward function, $0 < \gamma < 1$ is the discount factor, S is set of all possible states, and P is the transition function. Note that if P is uniform over S at every state regardless of action, there is no need for planning at all: in a purely stochastic environment, planning is impossible. This holds true even if for a given state s there are a limited subset of states $S_{s\rightarrow} \subset S$ that have non-zero probability under P, because $P(s'|s,a_1) = P(s'|s,a_2)$ in all of these cases, i.e., the action an agent chooses to take has no effect on the outcome, so planning is entirely unnecessary.

On the other extreme, if an agent can freely choose their next state, e.g., $\forall s \in S, \forall s' \in S_{s\rightarrow}, \exists a_{s'} : P(s'|s,a_{s'}) = 1$, where $S_{s\rightarrow}$ is the set of reachable states from s, it will always do as well or better than an environment with the same connectivity but increased stochasticity:

$$\max_{a' \in A} \sum_{s' \in S} P(s'|s,a) R(s',a') \leq \max_{a' \in A} R(s',a') \tag{2}$$

Holding all else equal, a stochastic transition function means that even an optimal agent will sometimes enter a suboptimal solution due to incomplete control over future states.

Base LLMs adapted with RLHF face a similar conundrum to a highly stochastic transition function: they are forced to learn with a highly stochastic initial policy $\pi_{\text{Base LLM}}$ in an environment with a huge space of states and non-smooth distribution of rewards. Models therefore may tend towards previously high-reward states (e.g. the anchor spans), as this is less risky than exploration. In the exponentially large space of possible strings it is unsurprising that models tend to converge towards anchor spans.

Adapting LLMs for long-form generation is incentivizes the use of anchor spans, in order to avoid entering states for which $\pi_{\text{Base LLM}}$ has high entropy and is therefore hard to predict. This becomes exponentially harder to avoid as the length of generated strings grows (Dziri et al., 2024). We suggest that models converge to anchor spans to lower-bound success, and likely do so when adapted from a vanilla next-token-predictor, even with very different adaptation methods.

6 Related Work

Catastrophic Forgetting Prior work examined models forgetting previously learned distributions. In most continual learning settings, where the model continues to train on new data, this is termed *catastrophic forgetting* (Kirkpatrick et al., 2017). This phenomenon has been observed in Language Models, and several mechanisms have been proposed to mitigate its effects (Chen et al., 2020; Xu et al., 2020; Vu et al., 2022). More recently, catastrophic forgetting has also been found to impact modern generative LLMs (Luo et al., 2023).

Distribution collapse of LLMs Distribution collapse is known to occur in models which have been trained on data distributions that include model generations (Shumailov et al., 2023). Such self-consuming models exhibit a degradation in generation quality as well as diversity (Briesch et al., 2023; Alemohammad et al., 2023). Unlike these works, which do not consider preference tuning objectives at all, we focus on collapse in RLHF models. Other works do consider RLHF models, but do not study the nature of this collapse, instead focusing exclusively on methods for alleviating the style of *mode collapse* which arises from the degenerative overfitting to an imperfect reward model (Perez et al., 2022; Go et al., 2023). These methods are taken from the RL literature (Jaques et al., 2017), and build on studies which find mode collapse common in many other models (Che et al., 2017; Mescheder et al., 2018). Xiao et al. (2024) also studies RLHF models, specifically on collapse at a *preference* level over all generations. Our work, on the other hand, visualizes and describes, quantitatively and qualitatively, the sequence-level multi-token repetition in generations after RLHF, as characterized by the presence of *anchor spans*.

7 Conclusion

We present evidence that: (1) RLHF-adapted LLMs are no longer performant next-token predictors, and thus no longer serve as world models of the textual space. (2) Such models collapse their distribution into a more predictable subdistribution of the base distribution. (3) RLHF LLMs *blueprint* long-form generations via anchor spans.

We argue that these differences represent a fundamental trade-off between world models and agent

models. An agent model that takes action via sampling must reshape its distribution in such a way that it no longer represents the full distribution of possibilities that world models capture, ensuring that it doesn't "go off the rails" as Base LLMs are prone to. Future work could explore strategies that can mitigate the observed trade-off between the predictive capabilities of world models and the action-oriented nature of agent models. For instance, a system that decides when to call a world model vs. an agent model may be able to more reliably switch between these for different goals, using the world model as a probabilistic simulator that helps the agent model decide how to act.

Limitations

One key limitation of our work that will need to be addressed in future literature is comprehensiveness across a broader range of models, as well as varied methods for agent-alignment. We focus on RLHF here as this is the most popular and successful method currently being used. We also aimed to test popular and performant models, as these represent the limits of both RLHF and base LM capabilities. In future work, it will be important to study more various models, including more model scales. It will also be useful to include multiple random seeds for training/tuning in all experiments, but the cost of such experiments would be infeasible here.

It will also be useful in future work to study the blueprints produced by RLHF models in more detail. This work aims to both demonstrate the trade-off between agent and world modeling, then explore different aspects of this blueprinting. However, dedicated work will be required to fully characterize this process.

Ethical Considerations

Aligned models, such as those tuned with RLHF, have seen a recent explosion in capabilities, popularity, and deployment compared to traditional LLMs trained primarily on broad text prediction. As part of this shift, aligned models are much more frequently framed as "agents" which can take explicit actions, and take active part in human-facing systems. This framing poses significant risks without a better conceptual understanding of these techniques, particularly with respect to whether these models are indeed robust agents and what underlying learned mechanisms might allow for this.

Our work seeks to contribute to this understanding. For example, we find some evidence that aligned models may indeed be planning, which supports a notion of "agentiveness". Yet we also find that these models lose robust and accessible notions of calibrated text prediction, which could indicate a tendency towards biased heuristics or at least away from the more robust, broad-text understanding of traditional LMs. Overall, our work indicates ongoing concerns with treating aligned models as well-informed agents, and demands further study into these aspects and risks.

References

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G Baraniuk. 2023. Self-consuming generative models go mad. *arXiv preprint arXiv:2307.01850*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Martin Briesch, Dominik Sobania, and Franz Rothlauf. 2023. Large language models suffer from their own output: An analysis of the self-consuming training loop. *arXiv preprint arXiv:2311.16822*.

Tong Che, Yanran Li, Athul Jacob, Yoshua Bengio, and Wenjie Li. 2017. Mode regularized generative adversarial networks. In *International Conference on Learning Representations*.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. *arXiv preprint arXiv:2004.12651*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. 2019. On the use of arxiv as a dataset. *Preprint*, arXiv:1905.00075.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. 2024. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. 2023. Aligning language models with preferences through f-divergence minimization. *Preprint*, arXiv:2302.08215.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. 2024. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Eduard Hovy and Chin-Yew Lin. 1998. Automated text summarization and the summarist system. In *TIPSTER TEXT PROGRAM PHASE III: Proceedings of a Workshop held at Baltimore, Maryland, October 13-15, 1998*, pages 197–214.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*.

Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E. Turner, and Douglas Eck. 2017. Sequence tutor: Conservative fine-tuning of sequence generation models with KL-control. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1645–1654. PMLR.

Kazutaka Katoh and Daron M Standley. 2013. Mafft multiple sequence alignment software version 7: improvements in performance and usability. *Molecular biology and evolution*, 30(4):772–780.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. 2024. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason E Weston, and Mike Lewis. 2023a. Self-alignment with instruction backtranslation. In *The Twelfth International Conference on Learning Representations*.

Yucheng Li, Frank Guerin, and Chenghua Lin. 2023b. Avoiding data contamination in language model evaluation: Dynamic test construction with latest materials. *Preprint*, arXiv:2312.12343.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. 2018. Which training methods for GANs do actually converge? In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3481–3490. PMLR.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C Wallace, and David Bau. 2023. Future lens: Anticipating subsequent tokens from a single hidden state. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 548–560.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernandez. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. *Preprint*, arXiv:2202.03286.

Adam Roberts, Colin Raffel, Katherine Lee, Michael Matena, Noam Shazeer, Peter J. Liu, Sharan Narang, Wei Li, and Yanqi Zhou. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. Technical report, Google.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant. 2022. Overcoming catastrophic forgetting in zero-shot cross-lingual generation. *arXiv preprint arXiv:2205.12647*.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. 2023. Mint: Evaluating llms in multi-turn interaction with tools and language feedback. In *The Twelfth International Conference on Learning Representations*.

Jiancong Xiao, Ziniu Li, Xingyu Xie, Emily Getzen, Cong Fang, Qi Long, and Weijie J. Su. 2024. On the algorithmic bias of aligning large language models with rlhf: Preference collapse and matching regularization. *Preprint*, arXiv:2405.16455.

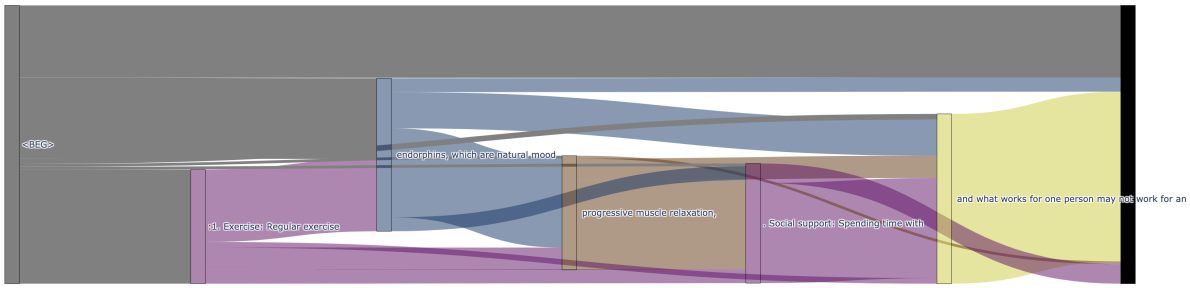
Ying Xu, Xu Zhong, Antonio Jose Jimeno Yepes, and Jey Han Lau. 2020. Forget me not: Reducing catastrophic forgetting for domain adaptation in reading comprehension. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

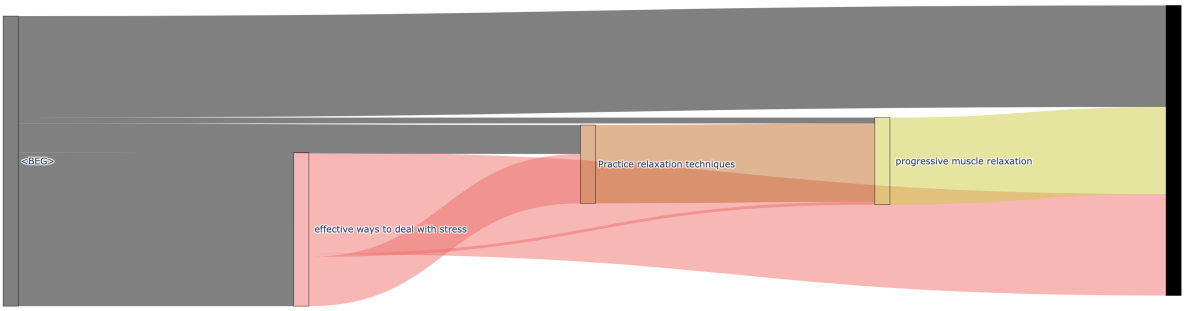
A Sankey Diagrams

Figures 9, 10, 11, 12 show further Sankey Diagrams—sampled at random from the 80 Vicuna prompts.



Llama 2 70B RLHF, Nucleus Sampling, p=0.9

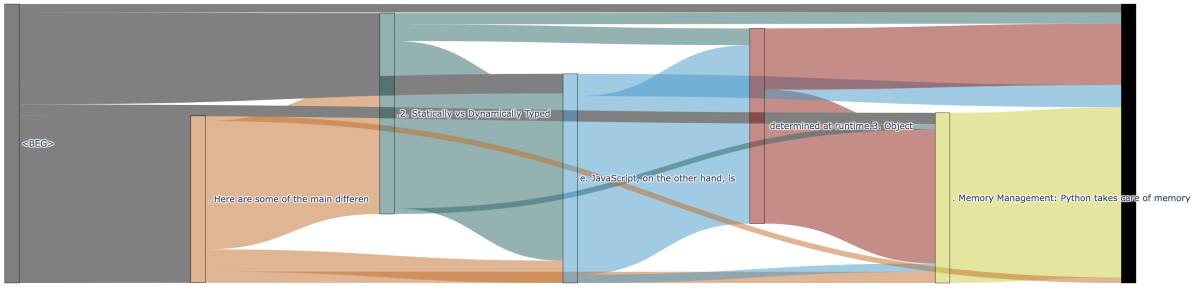
Llama 2 70B Base, Nucleus Sampling, p=0.7



Llama 2 7B RLHF, Nucleus Sampling, p=0.9

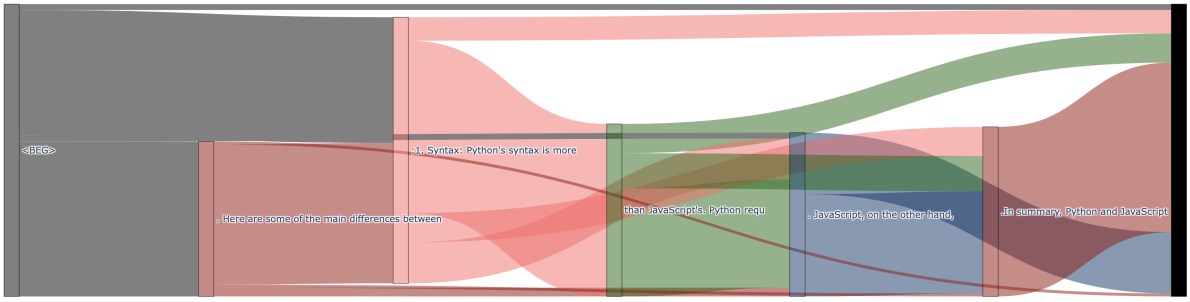
Llama 2 7B Base, Nucleus Sampling, p=0.7

Figure 9: (§4.2) Further examples of Sankey diagrams, as in Figure 8. Each Sankey diagram visualizes 100 responses (nucleus sampling, p = 0.9 for RLHF and p = 0.7 for Base) to the prompt "What are the most effective ways to deal with stress?"



Llama 2 70B RLHF, Nucleus Sampling, p=0.9

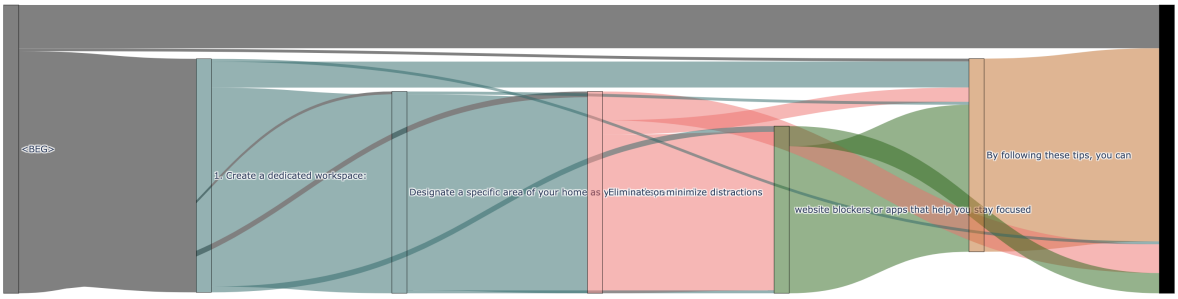
Llama 2 70B Base, Nucleus Sampling, p=0.7



Llama 2 7B RLHF, Nucleus Sampling, p=0.9

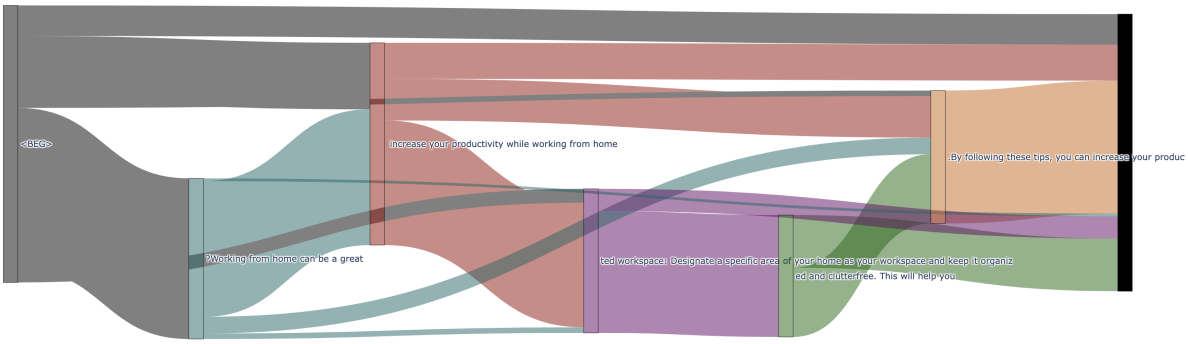
Llama 2 7B Base, Nucleus Sampling, p=0.7

Figure 10: (§4.2) Further examples of Sankey diagrams, as in Figure 8. Each Sankey diagram visualizes 100 responses (nucleus sampling, p = 0.9 for RLHF and p = 0.7 for Base) to the prompt "What are the main differences between Python and JavaScript programming languages?"



Llama 2 70B RLHF, Nucleus Sampling, p=0.9

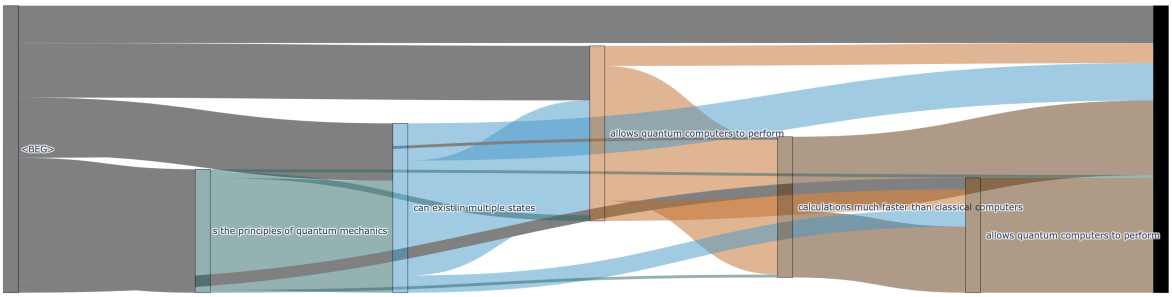
Llama 2 70B Base, Nucleus Sampling, p=0.7



Llama 2 7B RLHF, Nucleus Sampling, p=0.9

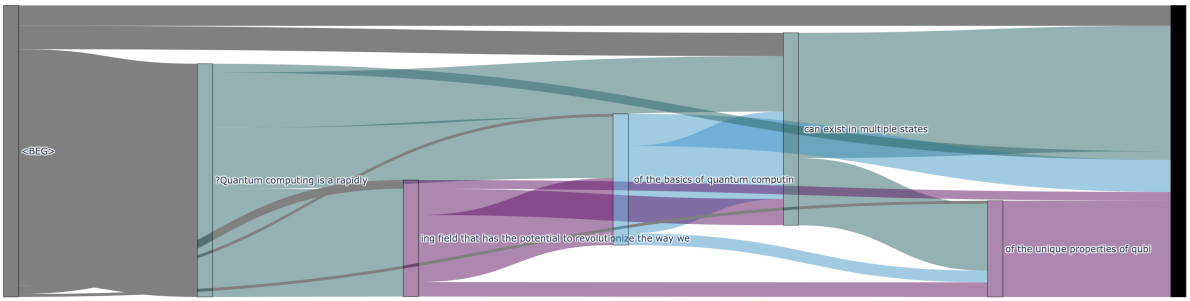
Llama 2 7B Base, Nucleus Sampling, p=0.7

Figure 11: (§4.2) Further examples of Sankey diagrams, as in Figure 8. Each Sankey diagram visualizes 100 responses (nucleus sampling, p = 0.9 for RLHF and p = 0.7 for Base) to the prompt "How can I increase my productivity while working from home?"



Llama 2 70B RLHF, Nucleus Sampling, p=0.9

Llama 2 70B Base, Nucleus Sampling, p=0.7



Llama 2 7B RLHF, Nucleus Sampling, p=0.9

Llama 2 7B Base, Nucleus Sampling, p=0.7

Figure 12: (§4.2) Further examples of Sankey diagrams, as in Figure 8. Each Sankey diagram visualizes 100 responses (nucleus sampling, p = 0.9 for RLHF and p = 0.7 for Base) to the prompt "Can you explain the basics of quantum computing?"

B Overlap Graphs

Figure 13 plots the same overlap metrics as Figure 1 (top), but for more models. Figure 14 shows that these metrics show a similar story when measuring the proportion of sequences that participate in *anchor spans* (see §4.2) rather than looking at string alignment directly.

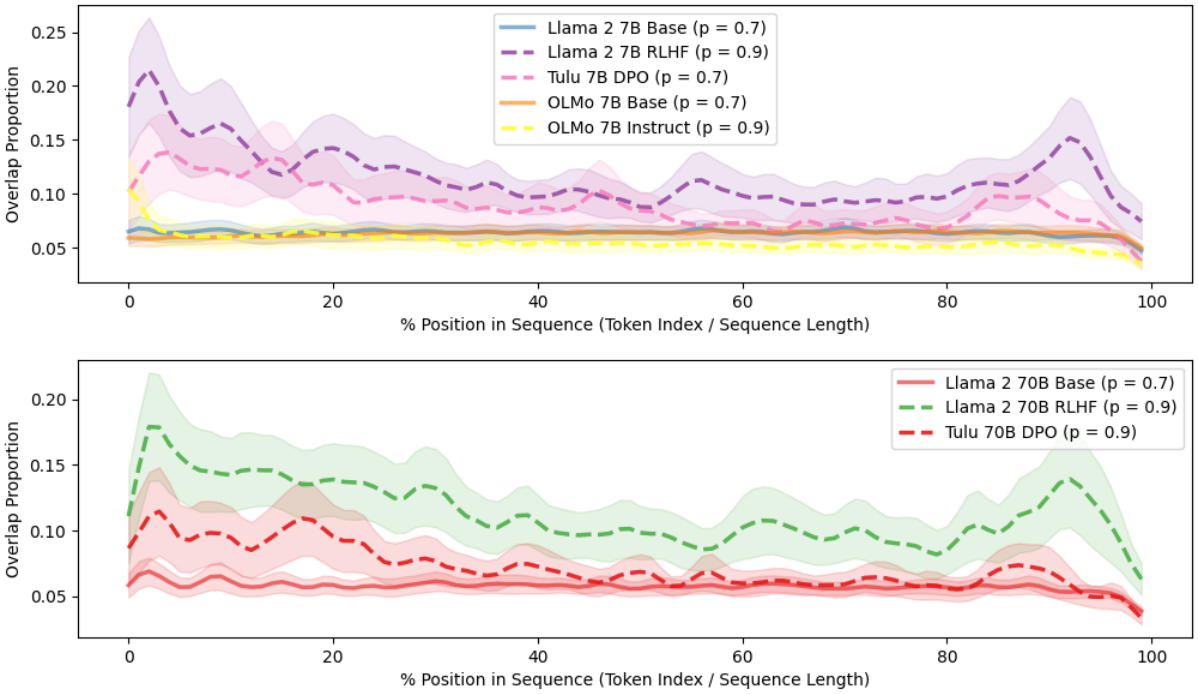


Figure 13: **RLHF model generations on the same prompt are highly similar to each other, unlike Base LMs.** For each of 80 short prompts, we collect and align 100 generations (nucleus sampling, p = 0.9) from Base (pretrained) and RLHF models. (§4.3) Over the sequence length, the number of generations aligned with at least 5 others, averaged over all prompts. Base model generations maintain low levels of alignment. RLHF model generations exhibit high alignment throughout. **Above:** (§4.2) LLMs with 7B parameters **Below:** LLMs with 70B parameters.

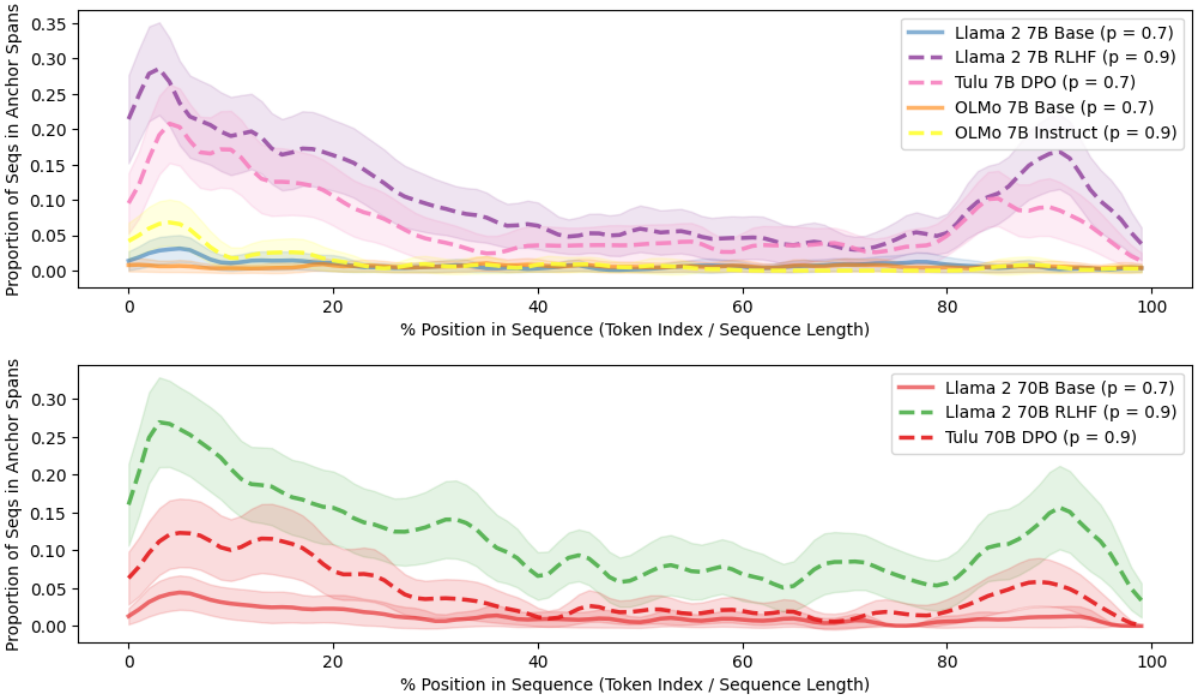
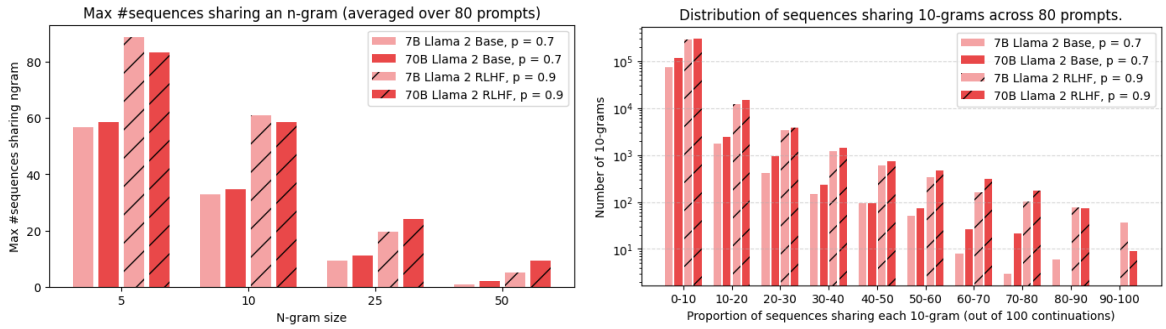


Figure 14: **RLHF model generations on the same prompt are highly similar to each other, unlike Base LMs.** For each of 80 short prompts, we collect and align 100 generations (nucleus sampling, p = 0.9) from Base (pretrained) and RLHF models. (§4.3) While Figures 1 and 14 look at alignment on raw characters, these diagrams show that when considering directly what proportion of sequences are part of an *anchor span* (see §4.2) the pattern remains. **Above:** (§4.2) LLMs with 7B parameters **Below:** LLMs with 70B parameters.

C Ngram Charts

Figure 15 shows the same statistics as Figure 7 for further models.



(A) **(B)**

Figure 15: **(§4.3) Given a short prompt, RLHF models heavily reuse n-grams across many independently generated continuations (nucleus sampling, p = 0.9), with the most common 10-gram appearing in 60% of generations on average.** For each of 80 short prompts, we collect 100 generations from Base and RLHF models using nucleus sampling with p = 0.7 and p = 0.9, respectively. **(A)** For each prompt, the number of generations, out of 100 total, which contain the most common n-gram ($n \in [5, 10, 25, 50]$), averaged across all prompts. **(B)** A histogram, binning 10-grams by the number of sequences containing that 10-gram. Counts are log-scale. Compared to Base models, RLHF models much more frequently generate the same 10-gram in nearly all continuations for a prompt.

D Shannon Game

What if the ranking information about which tokens are more or less likely is still preserved in RLHF models, even if the probabilities themselves are distorted? The perplexities RLHF adapted LLMs yield are higher, but this could be potentially be a result of the distribution collapse RLHF models undergo (see §3). To evaluate how well LLMs rank the gold next token, we evaluate both Base and RLHF models using the Shannon Game.

Experimental setup The Shannon Game (Hovy and Lin, 1998), is a next token prediction task, except no probabilities are used. Instead, models are judged by the number of *incorrect guesses* that a model ranks with a higher *score* over the target token. The Shannon Game is invariant to relative differences in exactly how much probability is allocated to different strings, and is only sensitive to the *ordering* that tokens are given in the hypothesis. We evaluate the Base Llama 2 and RLHF Llama 2 (the Chat version) on LAMBADA (Paperno et al., 2016), a collection of narrative passages designed to test the ability of LLMs on predicting the final word of a whole passage.

Results Table 1 shows that this is not the case via the Shannon Game, revealing that RLHF adapted LLMs are worse at ranking possible next-tokens, not just assigning them probability. Table 1 shows that RLHF models are worse at the Shannon Game, suggesting that the ability to model arbitrary aspects of the textual world are diminished by current agent adaptation techniques.

While it is tempting to assume that this is merely a result of imperfect RLHF methods, we argue that this trade-off is inherent to agent-adaptation. To generate multiple hundreds of tokens towards a singular goal, an agent model must limit the amount of *uncertainty about future tokens*. Planning a coherent document while marginalizing over all possible paths is an *exponentially* harder problem then collapsing onto a small set of possibilities. We hypothesize that the long-form generation capabilities of current RLHF models, are a general feature agent models: limiting the subspace of possibilities for any given prompt allows for better planning within this subspace. Further evidence for this hypothesis is given in §4.

Llama 2	Condition	EM	F1	Avg. guesses
7B	Base	68.41	70.31	7.62
	RLHF	56.70	60.78	9.99
70B	Base	67.60	73.10	6.18
	RLHF	66.47	67.50	8.64

Table 1: **RLHF models are worse than Base models at ranking possible next tokens on the LAMBADA dataset** (Paperno et al., 2016), requiring more incorrect guesses to identify the correct token (Avg. guesses).

E Experimental Setups

E.1 Perplexity Datasets

We list the corpora used in our perplexity experiments (§2) in Table 2.

Table 2: Data used for perplexity experiments in §2

Category	Data
Pretraining	Wikipedia C4 (Roberts et al., 2019) Arxiv (Clement et al., 2019)
New Corpus	New BBC (Li et al., 2023b) New Arxiv (Li et al., 2023b)
Instruction Data	Humpback (Li et al., 2023a)
Chat	Anthropic Harmless (Bai et al., 2022) Anthropic Helpful (Bai et al., 2022) OASST1 (Köpf et al., 2024)

E.2 Self-Perplexity Datasets

We list the corpora used in our self-perplexity experiments (§3) in Table 3.

Table 3: Data used for self-perplexity experiments in §3

Category	Data
Pretraining	Wikipedia C4 (Roberts et al., 2019) Arxiv (Clement et al., 2019)
New Corpus	New BBC (Li et al., 2023b) New Arxiv (Li et al., 2023b)
Instruction Data	Humpback (Li et al., 2023a) LIMA (Zhou et al., 2023)
Chat	Anthropic Harmless (Bai et al., 2022) Anthropic Helpful (Bai et al., 2022) OASST1 (Köpf et al., 2024) Vicuna (Chiang et al., 2023)

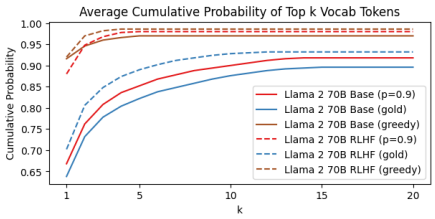


Figure 16: **§3 RLHF models assign nearly all of the next-token probability mass to a single token, more than Base models.** For Base and RLHF models, we calculate the next-token probability distributions on the gold sequences, as well as on the models' own generations (nucleus sampling, p=0.9; and greedy). We show the cumulative probability mass of the tokens, sorted in descending order of probability. RLHF models assign a larger portion of the probability mass to a very small number of tokens, compared to Base models.

E.3 Setup for distribution collapse experiments

To demonstrate agent model collapse, we checked cumulative probability of the most probable k tokens (Figure 3) and percentage of non-negligible vocabulary (Figure 4) for RLHF and Base models on gold vs. their generations, averaged across the diverse dataset used in Table 3.

E.4 Planning experiments

We use the Vicuna dataset (Chiang et al., 2023) as a source of prompts and sample 100 continuations for each of the prompts from both 7B and 70B Base and RLHF models.

Diversity of *n*grams vs. nucleus sampling p. We sample 100 continuations for each of the 80 prompts of the Vicuna dataset (Chiang et al., 2023) and measure the proportion of unique *n*grams (at the token level) for different values of nucleus sampling p. We find that setting nucleus sampling $p = 0.7$ for Base models achieves a similar diversity of *n*grams as setting $p = 0.9$ for RLHF models. We report results controlled for *n*gram diversity by using these nucleus sampling values. Note that when using the same nucleus sampling values, the differences described are even more significant.

Alignment setup. We employ the sequence alignment software MAFFT (Multiple Alignment using Fast Fourier Transform) (Katoh and Standley, 2013) to align the 100 continuations of each prompt of the Vicuna dataset. The MAFFT software has a "text" setting designed to align multiple sequences with arbitrary characters. This is a method used in bioinformatics to align three or more biological sequences (generally protein, DNA, or RNA). In simple terms, it's a way of lining up these sequences to identify regions of similarity. The problem of aligning multiple generations is not trivial and, like in biological sequences, needs to handle deletions, insertions, substitutions, and translocations of tokens in the sequence. Biological alignment software is designed to handle these operations. In fact, we find MAFFT to be effective at aligning generations which might deviate at points due to some phrases being worded differently or omitted still often converge back later.

The plot in Figure 1 was obtained by applying MAFFT to the 100 generations for each prompt. We removed any position in the aligned output that was not shared by at least 5 sequences. We then calculate overlap by measuring the proportion of pairwise matches across sequences for each position. Next we downsampled the resulting sequence of overlap scores, which have varying lengths, to 100-dimensional sequence averaging corresponding values. Finally, we averaged across all prompts and applied a 1D Gaussian filter to smooth the curves. The shaded areas indicate confidence intervals for each position.

Model	Llama 2 70B					Llama 2 70B RLHF				
ngrams	p=0.6	**p=0.7**	p=0.8	p=0.9	p=1.0	p=0.6	p=0.7	p=0.8	**p=0.9**	p=1.0
1	4.11%	**5.12%**	6.73%	9.20%	15.69%	3.35%	3.63%	3.89%	**4.33%**	5.39%
2	16.45%	**21.34%**	29.34%	41.04%	60.61%	13.18%	14.70%	16.00%	**17.96%**	22.59%
3	29.10%	**37.56%**	50.77%	68.09%	86.80%	23.67%	26.73%	29.24%	**32.71%**	40.36%
4	38.33%	**48.54%**	63.34%	80.78%	94.22%	32.03%	36.31%	39.77%	**44.17%**	53.23%
Model	Llama 2 7B					Llama 2 7B RLHF				
ngrams	p=0.6	**p=0.7**	p=0.8	p=0.9	p=1.0	p=0.6	p=0.7	p=0.8	**p=0.9**	p=1.0
1	3.11%	**4.15%**	6.08%	9.34%	17.17%	3.06%	3.29%	3.67%	**4.03%**	5.42%
2	12.18%	**17.29%**	26.83%	42.26%	64.11%	12.51%	13.78%	15.67%	**17.29%**	23.08%
3	21.57%	**30.66%**	47.06%	70.26%	89.50%	23.43%	26.03%	29.64%	**32.65%**	42.14%
4	28.50%	**39.79%**	59.04%	82.79%	95.86%	32.62%	36.32%	41.21%	**45.14%**	56.26%

Table 4: Proportion of unique *n*grams as a function of nucleus sampling p.