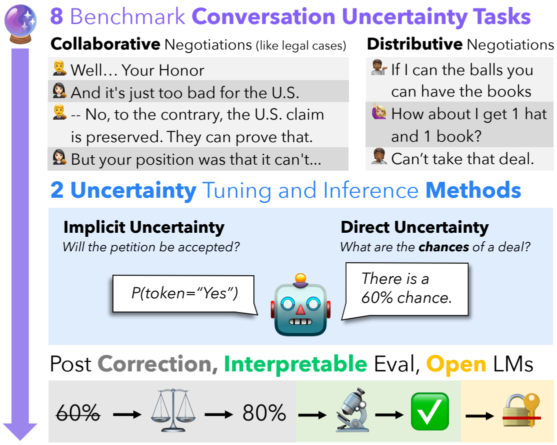
Deal, or no deal (or who knows)? Forecasting Uncertainty in Conversations using Large Language Models

Anthony Sicilia[♭] **Hyunwoo Kim**[♮] **Khyathi Raghavi Chandu**[♮]
Malihe Alikhani[♭] **Jack Hessel**[♯]
[♭]Northeastern University [♮]Allen Institute for AI [♯]Samaya AI
`{sicilia.a, m.alikhani}@northeastern.edu`
`{hyunwook, khyathic}@allenai.org` `jmhessel@gmail.com`

Abstract

Effective interlocutors account for the uncertain goals, beliefs, and emotions of others. But even the best human conversationalist cannot perfectly anticipate the trajectory of a dialogue. How well can language models represent inherent uncertainty in conversations? We propose **`FortUne Dial`**, an expansion of the long-standing "conversation forecasting" task: instead of just accuracy, evaluation is conducted with uncertainty-aware metrics, effectively enabling abstention on individual instances. We study two ways in which language models potentially represent outcome uncertainty (internally, using scores and directly, using tokens) and propose fine-tuning strategies to improve calibration of both representations. Experiments on eight difficult negotiation corpora demonstrate that our proposed fine-tuning strategies (a traditional supervision strategy and an off-policy reinforcement learning strategy) can calibrate smaller open-source models to compete with pre-trained models 10x their size.



8 Benchmark Conversation Uncertainty Tasks
Collaborative Negotiations (like legal cases)
Well… Your Honor
And it's just too bad for the U.S.
-- No, to the contrary, the U.S. claim is preserved. They can prove that.
But your position was that it can't…
Distributive Negotiations
If I can the balls you can have the books
How about I get 1 hat and 1 book?
Can't take that deal.
2 Uncertainty Tuning and Inference Methods
Implicit Uncertainty
Will the petition be accepted?
P(token="Yes")
Direct Uncertainty
What are the chances of a deal?
There is a 60% chance.
Post Correction, Interpretable Eval, Open LMs
~~60%~~
80%

Figure 1: **`FortUne Dial`** tests the ability of language models to represent uncertainty about future conversation outcomes. To meet this task, we tune models to express uncertainty directly in their output tokens or implicitly in their score distributions. We also provide additional strategies to correct uncertainty at inference-time. We propose tasks across 8 existing datasets, experimenting with GPT-4, Llama-2, and Zephyr-style models to release our best performing models publicly.

1 Introduction

Dialogue models are increasingly fluent, topical, and informative conversationalists, capable of predicting plausible next-utterances given a partial conversation. Yet, the capacity to generate a single, plausible utterance is not the same as modeling the *uncertainty* about all possible next-utterances in a calibrated way – that is, assigning an appropriate probability to potential conversation outcomes, reflective of the randomness we observe in the real world. For example, in negotiations, "Sounds good!" or "No thanks" may be equally fluent/topical/informative next-utterances, but one choice may be more likely if the *goals*, *beliefs*, and *emotions* of the interlocutors are taken into account. While even the best conversationalists cannot perfectly predict the trajectory of a dialogue, humans often manage uncertainty about social cues appropriately (Druckman and Olekalns, 2008), and demonstrate ability to both anticipate and affect the *likelihood* of future conversation outcomes (Ho et al., 2022). Meanwhile, it is not yet clear if language models posses even the simplest of these capabilities: *anticipation of outcome certainty*.

To study this, we expand the long-standing "conversation forecasting" task (Sokolova et al., 2008; Zhang et al., 2018). While the usual goal is to predict the outcome of an unfolding dialogue, we instead account for how well language models represent *uncertainty* about outcomes by measuring performance with calibration metrics. In effect, these calibration metrics allow models to abstain from predicting on instances when they estimate high uncertainty. Potential applications of models performant in this setting include: improved tools for studying the effects of strategy and social structure in negotiations (Curhan and Pentland, 2007), intervening to improve human and machine con-

Findings of the Association for Computational Linguistics: ACL 2024, pages 11700–11726
August 11-16, 2024 ©2024 Association for Computational Linguistics

versations (Lewis et al., 2017; Zhou et al., 2019; Schluger et al., 2022; Argyle et al., 2023), or assessing trust/heterogeneity in a data source via metrics like entropy (Csáky et al., 2019; Kuhn et al., 2022).

Here, we focus on the case of negotiations; this type of conversation is not only particularly sensitive to social uncertainties, but also, outcomes are readily quantified *post-hoc*. We ask language models questions about the likelihood of *deals*, *decisions*, and *emotional conflicts* in settings like *marketplaces*, *online forums*, and *courtrooms*, totaling 8 tasks to test uncertainty modeling in negotiations. Our contributions include:

1. formalizing the conversation uncertainty modeling task, along with its metrics (§ 2.1);
2. introducing two methods for representing uncertainty about the outcome of conversations using language models (§ 2.2);
3. and proposing fine-tuning (§ 2.3, § 2.4) and inference-time strategies (§ 2.5) for improving these representations.

We call this task 🔮**`FortUne Dial`**.[1] Experiments (§ 3) show GPT-4 and other large models can anticipate outcome certainty well, improving over prior knowledge by up to 9%. Moreover, results show the utility of our fine-tuning strategies: smaller (7B) models are tuned to outperform pre-trained, open-source models 10x their size. Indeed, metrics improve up to 11% on the tuning datasets and up to 3% out-of-distribution. Besides the performance of our model deliverables, experiments also communicate insight on the biases of pre-trained language models at this task, the ability of different models to make use of prior knowledge, and the generalization of different algorithmic strategies. Models and code are available on github.[2]

2 Modeling Uncertainty in Conversations

2.1 Problem, Notation, and Evaluation

Consider a natural language token set $\mathcal{T}$. We observe partial multi-party dialogues $D \in \mathcal{T}^*$ consisting of $K \sim \mathcal{U}\{2, L\}$ turns, with $L + 1$ being the eventual (random) length of the full dialogue.[3] Speaker turns are delimited by special sequences of tokens; e.g., “Speaker 4: ...” These partial conversations are unfinished, but have eventual outcome $O \in \mathcal{O} = \{0, 1\}$.[4] Nature picks a ***conversation distribution*** $\mathbb{D}$ over $\mathcal{T}^* \times \mathcal{O}$ which governs our supervised observations: $(D, O) \sim \mathbb{D}$. A ***forecaster*** f maps $D \mapsto \hat{P} \in [0, 1]$ where $\hat{P}$ estimates the probability $O = 1$.

Evaluation with Proper Scores A ***calibrated*** forecaster satisfies (Bröcker, 2009):

$$\mathbf{E}[O \mid \hat{P} = p] = p \quad \forall p \in \{f(x) \mid x \in \mathcal{T}^*\}, \tag{1}$$

which intuitively means if we consider all conversations assigned p by the forecaster, the mean occurrence of the outcome should also be p. While commonly used to asses the verity of general probability estimates (Guo et al., 2017) the constraint in Eq. (1) is often too broad because calibration, by itself, fails to measure the *variance* of a forecast (Ovadia et al., 2019). For example, the constant forecast $\hat{P} = \mathbf{E}[O]$ is calibrated, but rarely captures the *true* outcome probability (conditioned on the conversation). The issue of variance is *especially* important in our setting, where social and temporal uncertainties make anticipation difficult; i.e., the basic, indiscernible prediction $\mathbf{E}[O]$ may be competitive. To accommodate calibration *and* variance, we consider the constraint

$$\hat{P} = P \overset{\text{def}}{=} \mathbf{E}[O \mid D] \tag{2}$$

One way to achieve this is by optimizing a ***scoring function*** $\mathsf{s} : [0, 1] \times \mathcal{O} \to \mathbb{R}_{\geq 0}$:

$$\min_f \mathbf{E}[\mathsf{s}(\hat{P}, O)]. \tag{3}$$

If the scoring function is ***strictly proper***,[5] Eq. (2) is satisfied by the minimizer of (3), so solving (3) recovers the true uncertainty as desired. Moreover, Eq. (3), indeed, optimizes variance and calibration equally, among other nice properties for ranking suboptimal forecasts (Bröcker, 2009).

Tangible Scores We use proper scores only, such as the **Brier Score** (**BS**; Brier, 1950), which is the mean squared error between forecast probabilities and true outcomes. While the use of proper scores is important (see previous), they do present some caveats: (1) they lack interpretable units and (2) for

[1] **For**ecas**t**ing **Un**c**e**rtainty in **Dial**ogue.

[2] https://github.com/anthonysicilia/fortune-dial

[3] Uncertainty may be higher at different points in the dialogue (e.g., the beginning). Uniform sampling ensures we capture all scenarios (both high and low uncertainty), evaluating the model on diverse contexts.

[4] We only consider binary outcomes, but are flexible in application of this formulation, e.g., we can ask multiple questions to handle more outcomes in a one-vs-all fashion.

[5] Strict propriety requires that $\mathbf{E}[\mathsf{s}(P, O)] \leq \mathbf{E}[\mathsf{s}(\hat{P}, O)]$ for all $\hat{P}$ with equality if and only if $\hat{P} = P$.

fixed tasks, they often vary on a small scale.[6] To resolve these issues, we sometimes focus evaluation on the **Brier skill score**:

$$\textbf{BSS} = 1 - \textbf{BS}/\textbf{BS}_{\texttt{ref}} \tag{4}$$

where **BS** is the Brier score of the forecaster we are evaluating and $\textbf{BS}_{\texttt{ref}}$ is the Brier score of some reference forecaster. One way to interpret the skill score is *the percent improvement* of the forecaster compared to the reference. A simple reference, first proposed by Brier (1950), is the constant prediction $\mathbf{E}[O]$, in which case $\textbf{BS}_{\texttt{ref}}$ happens to be the *variance* of the outcome. Here, we may interpret the skill score as *the percent of variance* in outcome that is explained by the forecaster, like an R^2-value. On the other hand, $\mathbf{E}[O]$ can also be viewed as *prior knowledge*, obtainable before observing D, implying skill conveys improvement over our prior knowledge. As desired, skill score tends to vary more than Brier score, while also having an interpretable unit (percentage).

Aleatoric and Epistemic Uncertainty There are traditionally two main types of uncertainty which are studied in machine learning (and other) literature: *epistemic uncertainty*, which relates to uncertainty caused by a person or model's knowledge (lack thereof) and *aleatoric uncertainty*, which relates to inherent uncertainty in the data and is independent of the human/model (Hüllermeier and Waegeman, 2021). Importantly, this work only aims to improve quantification of the aleatoric uncertainty (inherent to the data) without consideration of the epistemic factors that impact *specific* interlocutors. We leave this for future research.

2.2 Language Models as General Forecasters

An (auto-regressive) language model $\texttt{LM}_\theta$ is a function parameterized by $\theta \in \mathbb{R}^d$ that returns a distribution over the next token $t \in \mathcal{T}$ conditional to any *prefix* $x \in \mathcal{T}^*$. We write $T \sim \texttt{LM}_\theta(x)$ for a *single token sample* and $T \sim \texttt{LM}^*_\theta(x)$ for the *iterated sampling process*, wherein we append a sampled token to x and re-sample until a stopping condition. We define a **prompt** Φ as a function $\Phi : \mathcal{T}^* \to \mathcal{T}^*$ such that for any input x, it holds that x is substring of $\Phi(x)$. So, Φ takes an input text x and modifies it to a new text $\Phi(x)$, which contains the original text and (usually) adds important meta-information for solving the task; e.g., goal descriptors, expected output, and other context. We consider two types of prompts, which can turn a language model into a probability forecaster:

1. **Implicit Forecasts** ($\texttt{IF}$): The prompt $\Phi_\mathcal{O}$ poses the question "Given the partial dialogue D, will the outcome represented by $\mathcal{O}$ occur?" Then, the language model forecasts as

$$\hat{P}_{\texttt{IF}} = \mathbf{P}\{T = \texttt{yes}\};\ T \sim \texttt{LM}_\theta \circ \Phi_\mathcal{O} \circ D \tag{5}$$

   where $\texttt{yes} \in \mathcal{T}$ is an affirmation token and $\circ$ is function composition.

2. **Direct Forecasts** ($\texttt{DF}$): The prompt $\Phi_\mathcal{O}$ poses the modified question "Given the partial dialogue D, *what is the probability* the outcome represented by $\mathcal{O}$ will occur?" Then, the model forecasts as:

$$\hat{P}_{\texttt{DF}} = \mathsf{p} \circ T; \quad T \sim \texttt{LM}^*_\theta \circ \Phi_\mathcal{O} \circ D \tag{6}$$

   where $\mathsf{p} : \mathcal{T}^* \to [0, 1]$ is a parser that extracts a "probability estimate" from sample T; i.e., the model answers directly in natural language.

More details on prompts are in § 3. Abstractly, both prompts describe the uncertainty modeling task using language, but make different assumptions.

2.3 Uncertainty Tuning of Implicit Forecasts

We consider a language model with pre-trained parameters θ_{init}, e.g., pre-tuned to follow instructions (Ouyang et al., 2022). The model computes a score vector $Z|\Phi(D) \in \mathbb{R}^{|\mathcal{T}|}$ and uses Z to forecast:

$$\hat{P}_{\texttt{IF}} = \mathbf{P}\{T = \texttt{yes}\} = \frac{\exp(Z_{\texttt{yes}}/\tau)}{\sum_{t \in \mathcal{T}} \exp(Z_t/\tau)} \tag{7}$$

where temperature τ is a fixed hyper-parameter. A fine-tuning objective can then be written:

$$\max_{\theta\,:\,\theta_{\text{init}} \to \theta} \mathbf{E}[O \ln \hat{P}_{\texttt{IF}} + \overline{O} \ln \mathbf{P}\{T = \texttt{no}\}] \tag{8}$$

where $\texttt{no}$ is a dis-affirmation token. In effect, Eq. (8) translates the objective in Eq. (3) to a fine-tuning objective by picking s to be the negative log score (a proper score, essentially equivalent to standard cross-entropy). Jiang et al. (2021) also consider calibration of pre-trained language models by direct supervision (as above), but focus on "factual" question-answering tasks where answers are more clearly right/wrong and the inherent social/temporal uncertainties of conversation are absent. In addition to a difference of setting, our

[6] This is perhaps the reason for common *improper* choices like calibration error, that do not account for forecast variance.

proposal also differs from Jiang et al. (2021) because we retain the language model's *whole* token distribution during inference, instead of only a candidate set. In § A.1, we provide a first theoretical and empirical characterization of the impact of this choice when fine-tuning language models. Our main observation is these techniques are practically equivalent at inference-time with less than 1% average difference in forecast (for our corpora). Thus, we advocate to retain the whole token distribution, since it is more easily coupled with other language modeling tasks; e.g., it doesn't require special machinery, like a loss with separate normalization.

Sampling Distribution In practice, we consider several negotiation datasets, defining distributions $\mathbb{D}_1 \ldots \mathbb{D}_\ell$ and prompts $\Phi_1 \ldots \Phi_\ell$. At test-time, if dataset diversity is sufficient, we expect the forecasts will generalize to new, *possibly unseen*, environments $\mathbb{D}_{\ell+1}$ and prompts $\Phi_{\ell+1}$. Formally, this setup is called *domain generalization* (Blanchard et al., 2011; Muandet et al., 2013) and, while many approaches to this problem exist, simply training on the balanced aggregate of all available domains often performs best in practice (Gulrajani and Lopez-Paz, 2020); we take this approach in § 3.

2.4 Uncertainty Tuning of Direct Forecasts

Current pre-training strategies may prime models to express uncertainty best directly, via their output tokens; e.g. this is observed when models express uncertainty about factual correctness in question-answering (Tian et al., 2023). Ideally, despite the different setting, fine-tuning can preserve and capitalize on this predisposition. One challenge is that direct forecasts make Eq. (3) non-differentiable, due to the parser. So, we formulate direct forecast tuning as a Markov Decision Process. We use reward $R = -\mathsf{s}(\mathsf{p} \circ T, O)$ with $T \sim \mathtt{LM}^*_\theta \circ \Phi \circ D$ and set s to the log score (see Eq. 8). In effect, the reward is the negative score of our forecaster. Then, the usual objective $J(\theta)$ of this Markov Decision Process is:

$$\max_{\theta\,:\,\theta_{\text{init}} \to \theta} \mathbf{E}[R] = -\min_{\theta\,:\,\theta_{\text{init}} \to \theta} \mathbf{E}[\mathsf{s}(\hat{P}_{\text{DF}}, O)]. \tag{9}$$

That is, we recover the original forecasting objective. While significant machinery has been developed for reward optimization (see Sutton and Barto, 2018) we apply policy gradient.

2.4.1 Policy Optimization

We focus on gradient-based policy optimization techniques, like REINFORCE (Williams, 1992) and PPO (Schulman et al., 2017). In particular, we derive an *off-policy* version of the policy-gradient theorem, specific to our forecasting task, which uses Monte Carlo samples to produce unbiased estimates of the gradient-updates for our optimization problem. The *off-policy* aspect is an important one. It means we can iteratively sample *any* policy (distribution) over our token space $\mathcal{T}$, and use these demonstrations to learn θ. Thus, while tuning, we can prioritize *exploration* vs. *exploitation* however we like, which can be an important factor for acting optimally in very general environments (Jiang et al., 2023), as is desired by our framework.

Off-Policy Policy Gradient For any random variable X, define μ_X as the mass function of X. Then, for any reference model $\mathtt{Ref} : \mathcal{T}^* \to \Delta(\mathcal{T})$:

$$\begin{aligned}\nabla_\theta \mathbf{E}[R] = \mathbf{E}\Big[&\mathsf{s}_{\tilde{T}} \cdot \tfrac{\mu_T(\tilde{T})}{\mu_{\tilde{T}}(\tilde{T})} \cdot \nabla_\theta \log \mu_T(\tilde{T})\Big]\\ \text{where} \quad & T \sim \mathtt{LM}^*_\theta \circ \Phi \circ D,\\ & \tilde{T} \sim \mathtt{Ref}^* \circ \Phi \circ D,\\ & \text{and } \mathsf{s}_{\tilde{T}} = -\mathsf{s}(\mathsf{p} \circ \tilde{T}, O).\end{aligned} \tag{10}$$

We derive this in § A.3. While other off-policy policy gradient techniques exist (Degris et al., 2012; Imani et al., 2018; Kallus and Uehara, 2020), the specifics of our problem allow us to make simplifying assumptions and yield a "simpler" and unbiased estimate of $\nabla_\theta \mathbf{E}[R]$ as the above.

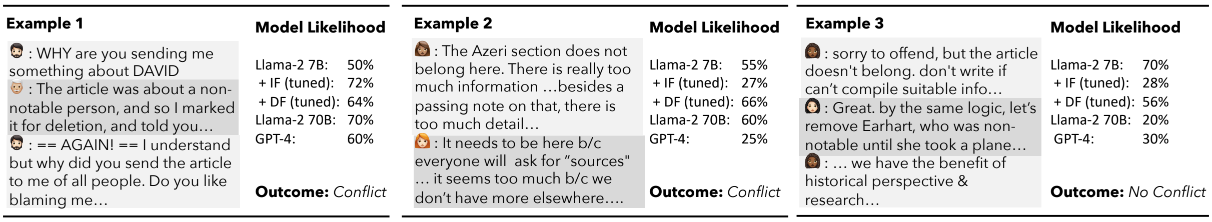
As a computational note, there may be instances where the ratio of mass functions $u_T / u_{\tilde{T}}$ becomes excessively large or small, leading to issues of gradient explosion or vanishing. To address this, we adopt a widely-used clipping strategy from Proximal Policy Optimization (Schulman et al., 2017). Specifically, for $\epsilon \in [0, 1]$, the update is:

$$\begin{aligned}&\mathsf{s}_{\tilde{T}} \cdot \omega \cdot \nabla_\theta \log \mu_T(\tilde{T}) \quad \text{where}\\ &\omega = \min\Big\{\max\Big\{\tfrac{\mu_T(\tilde{T})}{\mu_{\tilde{T}}(\tilde{T})}, 1-\epsilon\Big\}, 1+\epsilon\Big\}.\end{aligned} \tag{11}$$

Besides the computational benefits, this has motivations related to on-policy trust-region optimization (Schulman et al., 2015), when $\mathtt{Ref} = \mathtt{LM}_\theta$.

Off-Policies We consider three main choices for the off-policy reference $\mathtt{Ref}$ in this work:

1. **The Explorer** 🚀 takes random actions in the token space $\mathcal{T}^*$, restricted only in the sense that it must describe a probability; e.g., "53%." Exploration can benefit generalization (Jiang et al., 2023), since it exposes the agent to more diverse state-action pairs at train-time.



Example 1

🧔 : WHY are you sending me something about DAVID

👴 : The article was about a non-notable person, and so I marked it for deletion, and told you…

🧔 : == AGAIN! == I understand but why did you send the article to me of all people. Do you like blaming me…

Model Likelihood

Llama-2 7B:	50%
+ IF (tuned):	72%
+ DF (tuned):	64%
Llama-2 70B:	70%
GPT-4:	60%

Outcome: *Conflict*

Example 2

👩🏾 : The Azeri section does not belong here. There is really too much information …besides a passing note on that, there is too much detail…

👩 : It needs to be here b/c everyone will ask for "sources" … it seems too much b/c we don't have more elsewhere….

Model Likelihood

Llama-2 7B:	55%
+ IF (tuned):	27%
+ DF (tuned):	66%
Llama-2 70B:	60%
GPT-4:	25%

Outcome: *Conflict*

Example 3

👩🏾 : sorry to offend, but the article doesn't belong. don't write if can't compile suitable info…

👩🏻 : Great. by the same logic, let's remove Earhart, who was non-notable until she took a plane…

👩🏾 : … we have the benefit of historical perspective & research…

Model Likelihood

Llama-2 7B:	70%
+ IF (tuned):	28%
+ DF (tuned):	56%
Llama-2 70B:	20%
GPT-4:	30%

Outcome: *No Conflict*

Figure 2: Examples of model forecasts for the eventual occurrence of a *personal attack*. Models receive priors from data (§ 2.5) without any forecast scaling. Tuning (§ 2.3, § 2.4) improves 7B parameter models and GPT-4 shows bias against conflict, compared to other models (§ 3). The nuances that lead to conflicts are not necessarily obvious.

2. **The Exploiter** 💰 takes the actions that are optimal based on experience. We use $\hat{P}_{\texttt{IF}}$ has a proxy, since it does optimize a proper scoring function (Eq. 8). While optimal on the training data, this may not be true for new conversation domains or outcomes.
3. **The Quantizer** 📊 takes actions learned from ground truth data by binning. It is inspired by Lin et al. (2022), who calibrate model uncertainty to factual correctness by binning subtasks, computing average correctness in each bin, and training the model to predict these with routine supervision. Lacking clear "subtasks", we propose a (new) more general strategy, using clustering to bin our data. In § 3, we also ablate our use of RL to optimize, instead of routine supervision.

Greater detail on these policies, including precise definitions and implementation are in § A.4.

2.5 Post-Hoc, Inference-Time Corrections

A New View on Old Tricks If validation data is available, temperature (τ) scaling (Guo et al., 2017) is common to correct the scale of implicit forecasts (Jiang et al., 2021; Kadavath et al., 2022). Yet, this helpful technique is not well-studied for direct forecasts because these are parsed from a discrete token sequence (they have no underlying latent scores to scale). To address this, we propose a *unified* correction strategy that is well-suited for both implicit and direct forecasts. We suggest estimation of the underlying latent scores to allow **post-hoc scaling** ⚖️ for any forecast style:

$$\begin{aligned} \hat{Z}_{\texttt{yes}} &\leftarrow \log \hat{P}/(1-\hat{P}) \\ \tilde{Z}_{\texttt{yes}} &\leftarrow \hat{Z}_{\texttt{yes}}/\tau - \beta \\ \hat{P}_{\texttt{new}} &\leftarrow 1/(1+\exp(-\tilde{Z}_{\texttt{yes}})) \end{aligned} \tag{12}$$

where τ relates to temperature as before and β is a bias correction term. By estimating the latent score as above, we can effectively "simulate" traditional (τ) scaling in such a way that it works for *both* implicit and direct forecasts. We argue this theoretically and compare our proposal to other correction techniques in § A.2, finding it theoretically equivalent, practically equivalent, or better in general. We use Eq. (12) as the primary correction in § 3.

A Bayesian View Use of prior knowledge is considered an important qualifier for generalization in some theories of machine learning (McAllester, 1998). Motivated by this, we explore the use of natural language priors as a type of inference-time correction; e.g., , "On average, this type of conversation ends with {x} about {y}%..." Additional details on these priors are provided next.

Correction is Not Always Zero-shot Quality corrections require data in new domains. Data (n ≈ 250) is generally used to learn τ, β in Eq. (12), so scaling is not zero-shot. For priors, data can also be used or, in practice, a "guess" can be made. To simulate a "guessed" prior, we use averages *outside* a 95% confidence interval of the data average (n=50). Intuitively, this means we would rarely estimate this average using data (across repeated experiments), which is akin to, or worse than, human guesswork. § 3 considers `data priors` learned from *data* and `bad priors` using a *high* and *low* simulated guess.[7] We consider only the second to be zero-shot, since it tests robustness to "guesses" we expect to rarely obtain via data.

3 Experiments

Data & Splits We consider 8 modeling tasks spanning both traditional (distributive) negotiations and collaborative negotiations (Chu-Carroll and Carberry, 1995); Table 1 summarizes the corpora. Tasks have diverse situations and outcomes, span multi-party/dyadic settings, and are both short/long.

[7] We also checked non-numeric priors, e.g., reminding all outcomes have non-zero likelihood; these were worse overall.

Dataset	Situation	Outcome	# Speak	# Turn	# Char	Aff.	Distr.
Zhang et al.	wikipedia editing	personal attack	> 2	6.2	2.5K	yes	no
He et al.	craigslist	best deal for buyer	= 2	9.8	720	no	yes
Chawla et al.	camp provisions	both camps happy	= 2	11.4	1.2K	yes	yes
Chang et al.	reddit	personal attack	> 2	5.3	3.2K	yes	no
Wang et al.	charity	donation occurs	= 2	20.6	2.2K	yes	no
Lewis et al.	item allocation	deal occurs	= 2	5.0	253	no	yes
Mayfield et al.	wikipedia editing	article deleted	> 2	8.6	2.2K	no	no
Chang et al.[a]	courtroom	petitioner wins	> 2	218	55K	no	no

Table 1: Forecasting tasks. We list setting, outcome of interest, number of speakers, average turn/character count, and whether the setting is distributive. We also note if affective reasoning (about emotions) is useful in forecasting. Data are grouped into 3 train-test splits (*easy*, *med.*, *hard*) to simulate generalization difficulty (see Tables 6, 7). [a]See also Danescu-Niculescu-Mizil et al. (2012) for courtroom data. Precise outcome definitions are in § B.1.

	post-hoc scaling ⚖			*no scaling* ⚖					*combined*	
	~~`prior`~~	`data prior`		~~`prior`~~	`data prior`		`bad prior`		`neg`	`all`
model	**BS** ↓	**BS** ↓	**BSS** ↑	**BS** ↓	**BS** ↓	**BSS** ↑	**BS** ↓	**BSS** ↑	**BI**	**BI**
`gpt-4 DF`	21.1	20.7	8.5	23.8	21.8	3.8	22.7	9.0	-11	-1.1
🦙 `70B IF`	22.9	23.1	-2.0	66.1	66.1	-182	66.1	-160	3.6	-49.4
🦙 `70B DF`	22	22.1	2.3	25.0	22.0	2.7	23.6	5.0	-5.6	-1.3

Table 2: Scores for large, pre-trained models with different access to prior knowledge. Use of post-hoc correction and data-dependent priors is not truly zero-shot. GPT-4 uses direct forecasting (`DF`), while Llama-2-chat 🦙 70B uses direct or implicit (`IF`). **BSS** = 0 corresponds to always forecasting the prior, e.g., for `data prior`, this is the corpus mean outcome. `bad` priors are defined in § 2.5. `DF` consistently improves upon prior knowledge (**BSS** > 0).

	in-domain		*pseudo OOD*		*zero-shot OOD*					*combined*	
	`all`		`all`		`easy`	`med`	`hard`	`all`		`neg`	`all`
model	**BS** ↓	**LSS** ↑	**BS**	**LSS**	**LSS**	**LSS**	**LSS**	**BS**	**LSS**	**BI**	**BI**
🦙 `7B`			22.7	-3.2	-10.9	-7.5	1.1	24.3	-5.1	-0.4	-2.7
↪ `IF`	21	5.3	22.4	-1.4	-12.4	2.2	-3.9	23.9	-3.7	0.3	0
↪ `DF` 💰	22.8	-3.2	22.9	-3.7	-14.1	-8.2	-8.3	25.3	-9.7	0.9	0.6
↪ `DF` 🚀	22.9	-3.9	22.9	-3.6	-9.3	-8.5	1.7	24.2	-4.9	0.8	-0.9
↪ `DF` 📊 `rl`	22.9	-3.7	22.8	-3.3	-10.7	-5.9	2.1	24	-4.5	3	-0.6
↪ `DF` 📊 ~~`rl`~~	22.9	-3.8	22.9	-3.7	-11.9	-6.9	2.7	24.1	-4.9	3.4	-0.9
↪ `IF`$^{\times 4}$	19.6	11.4	21.4	3.7	8.9	2.9	-6.9	23.3	0.7	-3.5	-2.4
↪ `DF`$^{\times 4}$ 📊	22.8	-3.2	22.8	-3.4	-10.7	-3.7	4.5	23.6	-2.3	3.7	-2.8
🪁 `7B`			22.5	-1.2	-9.9	-9.6	-8.1	25.5	-9.1	-10.6	1.8
↪ `IF`	22.5	-1.6	22.8	-3.4	-26.3	-10.2	-6.8	25.8	-13	1.5	-9.1
↪ `DF` 📊 `rl`	23	-4.2	23	-4.2	-11.4	-8.9	-3.8	24.9	-7.6	0.2	-6.8
🪁 `1B`			22.8	-3.3	-11.2	-3	-2.6	24.4	-4.9	3.5	6.5
↪ `IF`	22.2	-0.5	22.8	-3.2	-19.4	-10.6	-3.2	25.2	-10	-2.4	-5.5
↪ `DF` 📊 `rl`	23	-4.2	23	-4.2	-11.4	-8.9	-3.8	24.9	-7.6	1	-1.2

Table 3: Scores for uncertainty tuned Llama-2-chat 🦙 7B, Zephyr 🪁 7B. , and "tiny" 1B Llama-2 trained in Zephyr style. First row (each section) provides a pre-tuned reference. *in-domain* shows test data scores from within tuning distribution, using val. data for post-hoc scaling and data priors. *pseudo OOD* and *zero-shot OOD* show scores when test data is out of distribution (i.e., held out domains), but only *pseudo OOD* uses scaling and data priors. *zero-shot OOD* doesn't scale and uses "bad" priors. For 🦙 7B, we also tune on ×4 more data. Improvements are highlighted: Light green cells show improvement against corresponding pre-trained models (same setup, before uncertainty tuning), while darker cells (additionally) improve over larger 🦙 70B DF (**LSS** *always* compares to DF).

To simulate varying degrees of distribution shift, we group these datasets into different train/test splits, categorized as `easy`, `medium`, or `hard`. To make the forecasting task more difficult, each of these three splits hold out full datasets for testing. Conceptually, the splits are designed to create different degrees of train/test imbalance for important properties like the topic, the length of the conversation, the type of outcome, and the number of speakers; Tables 6+7 and § B.1 provide more detail on train/val/test splits. Sometimes, we restrict inference to data with *affective conflict* as outcome (`neg`) like, a personal attack or unhappy speaker. Here, to compute some metrics, we swap the positive and negative classes as needed, e.g., the positive class becomes $1 - O$ to study "both camps unhappy" instead of "both camps happy."

Models & Prompts We use GPT-4 (0613, OpenAI, 2023), Llama-2-chat (Touvron et al., 2023), and Zephyr-β (Tunstall et al., 2023), which have all been pre-tuned for chat/instruction following (Zephyr is pre-tuned via distillation). We also use a "tiny" Llama-2 replicate trained in the style of Zephyr (Chat-v0.6, Zhang et al., 2023). Open-source model sizes range from 7B to 70B parameters. We use *pre-trained* to refer to these models before we tune uncertainty (§ 2.3, 2.4). Concretely, the prompts the models receive have: *situational context* specific to each dataset, like "the speakers are defending their opinions on an issue"; *priors*, as in § 2.5; and the *main question* that asks the model about the likelihood of outcome occurrence in the conversation, or just occurrence for implicit forecasts. Pre-trained models also receive system prompts to constrain output and clarify the task goals. We use QLoRA (Dettmers et al., 2023) for uncertainty tuning. Additional details are in § B.2.

Metrics We use the Brier Score (**BS**) and skill score (**BSS**) as discussed in § 2.1, macro-averaged across datasets and prompts. **BSS** refers to the original skill score (Brier, 1950) where the reference model in the skill score is the constant (average) outcome probability. When a prior is provided, we substitute this prior for the data mean in the reference score to account for how priors can implicitly bias forecasts. Indeed, variance around an (incorrect) prior is always higher than the true variance, so the prior-adjusted **BSS** reports the percent of this larger variance explained by the forecaster. Besides **BSS**, we also suggest a *new* skill score called the **Llama Skill Score** (**LSS**). **LSS** is identical to usual skill scores, i.e., Eq. (4), but uses the Brier score of Llama-2-chat 🦙 70B (direct forecasts, same experimental setup) as the reference score $\mathbf{BS}_{\mathtt{ref}}$. This quantifies how smaller fine-tuned models compare to this large model by % improvement. Since tuned models are smaller, % improvement is sometimes negative: a less negative value is a smaller decrease (compared to Llama 70B) which means better performance. Finally, we report *statistical bias* (**BI**) to convey average over- or under-estimation of outcome probability (positive or negative values, respectively). In addition to discussion (§ 1, 2.1), empirical motivation for uncertainty-aware metrics is given in Table 8, comparing **BSS** to more typical classification metrics.

3.1 Results and Discussion

Forecasting is Better with Correction Table 2 shows Brier and skill scores of pre-trained models without uncertainty tuning. We modulate priors and ablate post-hoc scaling. GPT-4 has lower (better) Brier scores when granted access to validation data to make corrections, e.g., via data priors or scaling. Scaling appears to have the greatest impact on scores as they are lowest, even when excluding the data prior. However, priors are still useful. In zero-shot settings (i.e., no scaling or data prior), GPT-4 performs better when it has access to "bad" guesses of prior probability, rather than no guess at all. Trends are similar for Llama-2-chat 70B.

How Priors Help Curiously, "bad" priors can also improve model inferences. We hypothesize the benefit of priors may also come from eliciting "chain-of-thought" behavior at inference-time (Wei et al., 2023), compounding the improvement gained from increased information access. Manual analysis of GPT-4 on the corpora of Zhang et al. (2018) shows that GPT-4 explains its answers 60-85% more frequently, when provided a prior (depending on exact prior setting).

Problems with Pre-trained Implicit Forecasts For Llama-2-chat 70B, we have access to scores of every token, so we can compare implicit forecasting to direct forecasting. In Table 2, implicit forecasting is worse for this pre-trained model. Echoing Kadavath et al. (2022), we find post-hoc correction is *vital* to improve pre-trained implicit forecasts. Moreover, degradation of uncorrected implicit forecasts is high, suggesting amplification of this effect in our unique (conversational) setting. Manual inspection suggests the model's logit probabilities for "Yes" tends to be much smaller than 1%. When scaled, these probabilities range more appropriately (for example, 10% to 90%) and become adequately predictive.

Have Data? Tuned Implicit Forecasts Are Best Based on the previous results, we focus on uncertainty tuning with access to a prior (even a "bad" one) and compare uncertainty tuned models to direct forecasts of pre-trained versions. We consider an *in-domain* setting first, wherein test data follows the tuning distribution, post-hoc correction is used, and priors are data-dependent. Here, Table 3 shows tuned implicit forecasts can significantly improve over pre-trained Brier score, even compared to models 10x their size. For instance, a tuned Llama-2-chat 7B improves over the 70B model by about 11% (or, 5% with less data). With enough training data, out-of-distribution (OOD) scores are also about 4% better than Llama-2-chat 70B scores (if both have access to data for correction).

Direct Tuning Generalizes Better (Sometimes) Next, we consider a zero-shot OOD setting *without* data for correction. Here, performance of tuned implicit forecasts is still good, for Llama-2-chat 7B. With enough data, tuning brings Llama-2-chat 7B implicit forecasts to the skill of its 70B counterpart (+0.7%), but for Zephyr-style (distillation-tuned) models, degradation is significant compared to (even) pre-tuning scores. In contrast, for all 7B models and different levels of data access, direct forecasts show consistent improvement of scores after uncertainty tuning (see light green cells). For Llama-2-chat 7B, tuned direct forecasts also handle difficult (`hard`) distribution shift up to 4% better than their 70B counterpart. As speculated earlier, direct forecast tuning may preserve some predispositions of pre-trained models to direct uncertainty signals; this may explain consistent generalization of these forecasts *beyond* the tuning distribution.

Qualitative Comparison of `IF` and `DF` Tuned implicit forecasts tend to have higher variance than tuned direct forecasts. Averaging over all tuning strategies in Table 3, implicit tuning of Llama-2-chat 7B leads to 2× the standard deviation in forecasts compared to direct tuning (about 11% and 5% SD, respectively). As noted in § 2.1, all else equal, a higher variance is preferred by our metrics in order to capture the discernability of forecasts. Potentially, directly tuned models tend towards distribution collapse due to insufficient regularization in our RL objective (Korbak et al., 2022). Methods to resolve this will be of interest in future work.

Impact of Scale Tuning of implicit forecasts allow in-domain scores of a 1B model to rival a model 70x it's size (-0.5%). But, tuning methods show less improvement, or even degradation, when applied to the 1B model OOD. Possibly, and especially since we use QLoRA tuning, the reduction in trainable parameters is detrimental to these methods. This, and the observed benefits of increasing data (see previous discussion), suggest our tuning techniques may follow neural scaling laws (Kaplan et al., 2020), meaning generalization is strongly dependent on model, data, and compute scale.

Exploration & Exploitation Llama-2-chat 7B findings indicate pure exploitation 💰 is detrimental to generalization when compared with pre-trained models (see absence of green scores). On the other hand, exploration 🚀 tends to offer some improvement, beating the pre-trained scores on `easy` and `all` as well as the 70B scores on `hard`. The best tuned direct forecasts use quantization 📊, which is neither completely random nor optimal on the train set, offering a balance of exploration/exploitation.

Benefits of RL For our best performing direct forecast tuning mechanism, we also ablate the role of using RL to tune; i.e., we use a traditional supervised update rule, similar[8] to Lin et al. (2022). We find improvements over pre-training are less consistent and worse overall.

Human Preference Tuning May Induce Bias We also postulate human preference tuning (RLHF; e.g., Ouyang et al., 2022) may bias pre-trained models to under-estimate negative affective conflicts, since these are presumably undesirable to human annotators. To study this, we report statistical bias (**BI**) of forecast probabilities in predicting negative (`neg`) emotional outcomes; i.e., personal attacks, unhappy interlocutors, or refusals to donate to charity. On average, direct forecasts from GPT-4 underestimate this probability by about 11%, while direct forecasts from Llama-2-chat 70B underestimate this probability by about 6%. For the same models, bias across `all` outcomes is not as staggering. Smaller language models do not necessarily exhibit this bias, but distillation-tuned models

[8] Ours is still more general, due to the clustering proposal.

may learn similar biases from their larger teachers (see 7B). Generally, uncertainty tuning does not appear to introduce as staggering bias against emotional conflict, but especially for distillation-tuned models, the overall bias can be elevated.

4 Related Works

Negotiation & Conversation Forecasting Negotiation and dispute modeling has a long history (Lambert and Carberry, 1992; Jameson et al., 1994; Traum et al., 2008; Lascarides and Asher, 2008) with early works hand-crafting models of interlocutor behavior by logical or discourse structures. Reinforcement learning in simulated environments offers improvement (Georgila and Traum, 2011; Efstathiou and Lemon, 2014) with most recent advances modeling opponents' dialogue acts (Keizer et al., 2017), word choices (He et al., 2018), and mental states (Yang et al., 2021; Chawla et al., 2022). Instead of full simulation, we focus on efficient and interpretable outcome models (Sokolova et al., 2008; Nouri and Traum, 2014). Outcome models, or forecasts, are also common in broader dialogue for proactive moderation of social media (Zhang et al., 2018; Kementchedjhieva and Søgaard, 2021) as well as predicting task-success (Walker et al., 2000; Reitter and Moore, 2007), mental health codes (Cao et al., 2019), emotions (Wang et al., 2020; Matero and Schwartz, 2020), situated actions (Lei et al., 2020), and financials (Koval et al., 2023). Among these, ours is first to propose and evaluate probabilistic methodology, modeling *dynamic uncertainty* for the first time. Our proposal is also *uniquely general*, operating independent of setting or outcome of interest. Indeed, we evaluate on general negotiations, looking beyond distributive applications (zero-sum games in specific markets) to include common *collaborative negotiations* (Chu-Carroll and Carberry, 1995), like planning, where parties share some goals, but conflicts arise from competing sub-goals or beliefs.

Language Models & Uncertainty Modern large language models fine-tuned to human preferences (Ouyang et al., 2022) are increasingly general, "unsupervised" multi-taskers. When queried on factual information, these models also represent uncertainty about their solutions with little to no supervision (Kadavath et al., 2022). Uncertainty (about correctness) has also been studied in smaller models without preference tuning (Desai and Durrett, 2020; Jiang et al., 2021; Dan and Roth, 2021) with many algorithms for improvement (Kong et al., 2020; Zhang et al., 2021; Li et al., 2022). Albeit similarly operationalized, modeling uncertainty about factual correctness is distinct from our focus in negotiations, which elicits modeling of social dynamics and mental states. Fewer works study uncertainty in social reasoning (Jiang et al., 2021; Hosseini and Caragea, 2022; Kumar, 2022). These look at smaller models without instruction-tuning, and lack focus on the interactive, temporal aspects of conversations that cultivate an inherent uncertainty about future outcomes. Ours is also one of few works that study how models communicate uncertainty directly via output tokens (Mielke et al., 2022; Tian et al., 2023), and fewer that propose tuning algorithms for this (Lin et al., 2022).

5 Conclusion

We show language models represent uncertainty about conversational outcomes quite well, depending on their size, inference strategy, training strategy, and access to prior knowledge. We design a task to evaluate this ability and show:

- large (commercial-scale) models do this well, provided limited data to pick hyper-parameters;
- without data, these models still offer improvement over low quality priors, like human guesses;
- specialized fine-tuning can elevate small open-source models to beat models 10x their size;
- current pre-trained language models may be predisposed to representing uncertainty in their textual outputs, instead of their logit distributions;
- exploration at train-time can be more beneficial for generalization (compared to exploitation);
- and finally, pre-trained models may be biased against forecasting emotional conflict.

This work (and task) presents a first step towards understanding how language models can anticipate the certainty of outcomes in interactive social situations. We make our code, models, and data open-source to promote continued research.

Limitations

While we explore a wide array of datasets and experimental setups, the generality of our conclusions are limited to what's explored in this paper. Further study, e.g., replication study with different data, models, and settings, would provide evidence to confirm the generalization of our findings. One aspect of particular interest, is the application of these techniques to languages other than English.

Indeed, there is evidence that the uncertainty representations of language models may experience performance degradation when applied to other languages (Ahuja et al., 2022; Krause et al., 2023), especially those which are low-resource.

Additionally, while our task is motivated by a desire to probe a language model's ability to anticipate social (un)certainties in conversation, there is no clear way for us to separate causation and correlation in this task. We cannot claim the language model actually "understands" the causes of social (un)certainties, like interlocutor mental states, since instead, it may be the case that language models capitalize on "superficial" or "spurious" statistical correlations associated with an outcome (Ho et al., 2022).

Finally, we quantify the quality of a forecast primarily through its improvement over prior knowledge. Effectively, this compares a forecaster with a performance lowerbound, demonstrating the forecaster is using information revealed through the dialogue to provide an improved prediction. In the future, a performance upperbound (such as human performance) would be useful to establish a ceiling for our goals. This is particularly important for proper scores, since it is exceedingly rare for forecasters to achieve a perfect value (e.g., 0 for the Brier score or 1 for the skill score).

Ethics Statement

The models we use and train may exhibit, or even amplify, biases contained in the training data, such as societal biases. Robustness to adversaries and natural token perturbations is also not guaranteed. In any application, ethical and safety considerations should be made, such as bias mitigation methodologies and careful human moderation.

Moreover, even with mitigation strategies in place, using these models can introduce unexpected biases into the ecosystems where they are deployed. Users of these models (e.g., for decision making) can very well be lead astray by our models' outputs, especially without appropriate scrutiny. The ramifications of such over-reliance can also be far reaching, impacting not only the model user, but many other downstream parties, at a scale which is amplified by the automation purposes our models serve. Deployment of these models should consider the potential impact on users, their potential over-reliance, and the far-reaching consequences un-checked use of our models can create.

Acknowledgements

We thank Yejin Choi for her helpful feedback. Jack Hessel partially completed this work while at AI2. Anthony Sicilia completed this work while on internship at AI2.

References

Kabir Ahuja, Sunayana Sitaram, Sandipan Dandapat, and Monojit Choudhury. 2022. On the calibration of massively multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4310–4323, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lisa P Argyle, Christopher A Bail, Ethan C Busby, Joshua R Gubler, Thomas Howe, Christopher Rytting, Taylor Sorensen, and David Wingate. 2023. Leveraging ai for democratic discourse: Chat interventions can improve online political conversations at scale. *Proceedings of the National Academy of Sciences*, 120(41):e2311627120.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. 2011. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24.

Glenn W Brier. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Jochen Bröcker. 2009. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 135(643):1512–1519.

Jie Cao, Michael Tanana, Zac Imel, Eric Poitras, David Atkins, and Vivek Srikumar. 2019. Observing dialogue in therapy: Categorizing and forecasting behavioral codes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5599–5611, Florence, Italy. Association for Computational Linguistics.

Jonathan P. Chang, Caleb Chiam, Liye Fu, Andrew Z. Wang, Justine Zhang, and Cristian Danescu-Niculescu-Mizil. 2020. Convokit: A toolkit for the analysis of conversations. In *SIGDIAL Conferences*.

Jonathan P. Chang, Cristian Danescu-Niculescu-Mizil, and . 2019. Trouble on the horizon: Forecasting the derailment of online conversations as they develop. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4743–4754, Hong Kong, China. Association for Computational Linguistics.

Kushal Chawla, Gale Lucas, Jonathan May, and Jonathan Gratch. 2022. Opponent modeling in negotiation dialogues by related data adaptation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 661–674, Seattle, United States. Association for Computational Linguistics.

Kushal Chawla, Jaysa Ramirez, Rene Clever, Gale Lucas, Jonathan May, and Jonathan Gratch. 2021. CaSiNo: A corpus of campsite negotiation dialogues for automatic negotiation systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3167–3185, Online. Association for Computational Linguistics.

Jennifer Chu-Carroll and Sandra Carberry. 1995. Response generation in collaborative negotiation. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 136–143, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Lucas Clarté, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborová. 2023. Expectation consistency for calibration of neural networks. *arXiv preprint arXiv:2303.02644*.

Richárd Csáky, Patrik Purgai, and Gábor Recski. 2019. Improving neural conversational models with entropy-based data filtering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5650–5669, Florence, Italy. Association for Computational Linguistics.

Jared R Curhan and Alex Pentland. 2007. Thin slices of negotiation: predicting outcomes from conversational dynamics within the first 5 minutes. *Journal of Applied Psychology*, 92(3):802.

Soham Dan and Dan Roth. 2021. On the effects of transformer size on in- and out-of-domain calibration. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2096–2101, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Cristian Danescu-Niculescu-Mizil, Lillian Lee, Bo Pang, and Jon Kleinberg. 2012. Echoes of power: Language effects and power differences in social interaction. In *Proceedings of the 21st international conference on World Wide Web*, pages 699–708.

Thomas Degris, Martha White, and Richard S Sutton. 2012. Off-policy actor-critic. In *International Conference on Machine Learning*.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Daniel Druckman and Mara Olekalns. 2008. Emotions in negotiation. *Group decision and negotiation*, 17:1–11.

Ioannis Efstathiou and Oliver Lemon. 2014. Learning non-cooperative dialogue behaviours. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 60–68.

Kallirroi Georgila and David Traum. 2011. Reinforcement learning of argumentation dialogue policies in negotiation. In *Twelfth Annual Conference of the International Speech Communication Association*.

Ishaan Gulrajani and David Lopez-Paz. 2020. In search of lost domain generalization. In *International Conference on Learning Representations*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, Brussels, Belgium. Association for Computational Linguistics.

Mark K Ho, Rebecca Saxe, and Fiery Cushman. 2022. Planning with theory of mind. *Trends in Cognitive Sciences*, 26(11):959–971.

Mahshid Hosseini and Cornelia Caragea. 2022. Calibrating student models for emotion-related tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9266–9278, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Eyke Hüllermeier and Willem Waegeman. 2021. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506.

Ehsan Imani, Eric Graves, and Martha White. 2018. An off-policy policy gradient theorem using emphatic weightings. *Advances in Neural Information Processing Systems*, 31.

Anthony Jameson, Bernhard Kipper, Alassane Ndiaye, Ralph Schäfer, Thomas Weis, and Detlev Zimmermann. 1994. Cooperating to be noncooperative: The dialog system PRACMA. In *KI-94: Advances in Artificial Intelligence, 18th Annual German Conference on Artificial Intelligence, Saarbrücken, Germany, September 18-23, 1994, Proceedings*, volume 861 of *Lecture Notes in Computer Science*, pages 106–117. Springer.

Yiding Jiang, J Zico Kolter, and Roberta Raileanu. 2023. On the importance of exploration for generalization in reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Nathan Kallus and Masatoshi Uehara. 2020. Statistically efficient off-policy policy gradients. In *International Conference on Machine Learning*, pages 5089–5100. PMLR.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Simon Keizer, Markus Guhe, Heriberto Cuayáhuitl, Ioannis Efstathiou, Klaus-Peter Engelbrecht, Mihai Dobre, Alex Lascarides, and Oliver Lemon. 2017. Evaluating persuasion strategies and deep reinforcement learning methods for negotiation dialogue agents. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 480–484, Valencia, Spain. Association for Computational Linguistics.

Yova Kementchedjhieva and Anders Søgaard. 2021. Dynamic forecasting of conversation derailment. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7915–7919.

Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. 2020. Calibrated language model fine-tuning for in- and out-of-distribution data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1326–1340, Online. Association for Computational Linguistics.

Tomasz Korbak, Ethan Perez, and Christopher Buckley. 2022. RL with KL penalties is better viewed as Bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1083–1091, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ross Koval, Nicholas Andrews, and Xifeng Yan. 2023. Forecasting earnings surprises from conference call transcripts. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8197–8209, Toronto, Canada. Association for Computational Linguistics.

Lea Krause, Wondimagegnhue Tufa, Selene Baez Santamaria, Angel Daza, Urja Khurana, and Piek Vossen. 2023. Confidently wrong: Exploring the calibration and expression of (un)certainty of large language models in a multilingual setting. In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 1–9, Prague, Czech Republic. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Sawan Kumar. 2022. Answer-level calibration for free-form multiple choice question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 665–679, Dublin, Ireland. Association for Computational Linguistics.

Lynn Lambert and Sandra Carberry. 1992. Modeling negotiation subdialogues. In *30th Annual Meeting of the Association for Computational Linguistics*, pages 193–200, Newark, Delaware, USA. Association for Computational Linguistics.

Alex Lascarides and Nicholas Asher. 2008. Agreement and disputes in dialogue. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, pages 29–36, Columbus, Ohio. Association for Computational Linguistics.

Jie Lei, Licheng Yu, Tamara Berg, and Mohit Bansal. 2020. What is more likely to happen next? video-and-language future event prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8769–8784, Online. Association for Computational Linguistics.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453, Copenhagen, Denmark. Association for Computational Linguistics.

Dongfang Li, Baotian Hu, and Qingcai Chen. 2022. Calibration meets explanation: A simple and effective approach for model confidence estimates. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2784, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Matthew Matero and H Andrew Schwartz. 2020. Autoregressive affective language forecasting: a self-supervised task. In *Proceedings of COLING. International Conference on Computational Linguistics*, volume 2020, page 2913. NIH Public Access.

Elijah Mayfield, Alan W. Black, and . 2019. Analyzing wikipedia deletion debates with a group decision-making forecast model. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW).

David A McAllester. 1998. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234.

Sabrina J Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. 2013. Domain generalization via invariant feature representation. In *International conference on machine learning*, pages 10–18. PMLR.

Elnaz Nouri and David Traum. 2014. Initiative taking in negotiation. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 186–193, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32.

John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

David Reitter and Johanna D. Moore. 2007. Predicting success in dialogue. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 808–815, Prague, Czech Republic. Association for Computational Linguistics.

Charlotte Schluger, Jonathan P Chang, Cristian Danescu-Niculescu-Mizil, and Karen Levy. 2022. Proactive moderation of online discussions: Existing practices and the potential for algorithmic support. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW2):1–27.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Marina Sokolova, Vivi Nastase, and Stan Szpakowicz. 2008. The telling tail: Signals of success in electronic negotiation texts. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

David Traum, William Swartout, Jonathan Gratch, and Stacy Marsella. 2008. A virtual human dialogue model for non-team interaction. *Recent trends in discourse and dialogue*, pages 45–67.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Marilyn Walker, Irene Langkilde, Jerry Wright, Allen L Gorin, and Diane Litman. 2000. Learning to predict problematic situations in a spoken dialogue system:

experiments with how may i help you? In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5635–5649, Florence, Italy. Association for Computational Linguistics.

Zhongqing Wang, Xiujun Zhu, Yue Zhang, Shoushan Li, and Guodong Zhou. 2020. Sentiment forecasting in dialog. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2448–2458, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.

Runzhe Yang, Jingxiao Chen, and Karthik Narasimhan. 2021. Improving dialog systems for negotiation with personality modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 681–693, Online. Association for Computational Linguistics.

Justine Zhang, Jonathan Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Dario Taraborelli, and Nithum Thain. 2018. Conversations gone awry: Detecting early signs of conversational failure. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1350–1361, Melbourne, Australia. Association for Computational Linguistics.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2023. Tinyllama.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. 2021. Knowing more about questions can help: Improving calibration in question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1958–1970, Online. Association for Computational Linguistics.

Yiheng Zhou, He He, Alan W Black, and Yulia Tsvetkov. 2019. A dynamic strategy coach for effective negotiation. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 367–378, Stockholm, Sweden. Association for Computational Linguistics.

A Theory and Derivations

A.1 Comparison of Implicit Forecasting Approaches in Inference and Uncertainty Tuning

When doing implicit forecasting with language models, we find two main approaches in the existing literature (i.e., on calibration to correctness). As in the main text, one can retain the whole token distribution during inference. Recall, this is written:

$$\hat{P}_{\texttt{IF}} = \frac{e^{Z_{\texttt{yes}}/\tau}}{\sum_{t \in \mathcal{T}} e^{Z_t/\tau}}. \tag{13}$$

On the other hand, one can consider a *normalized* probability, restricted to a set of candidate answers. For example, Jiang et al. (2021) suggest this approach for smaller models, lacking appropriate instruction following capability. In our context, the approach is described:

$$\hat{P}_{\texttt{IFN}} = \frac{e^{Z_{\texttt{yes}}/\tau}}{e^{Z_{\texttt{yes}}/\tau} + e^{Z_{\texttt{no}}/\tau}}. \tag{14}$$

Primarily, Jiang et al. (2021) argue for this strategy to cope with the spreading of probability across the many different ways to indicate "yes" or "no", dependent on the question. In modern instruction-following models tuned with human feedback, Kadavath et al. (2022) consider both approaches. In their notation, "P(True)" takes the former approach (using the whole distribution), while "P(IK)" tunes a classification head on top of the model's internal feature representation, making "P(IK)" equivalent to tuning in the fashion of Jiang et al. (2021). As we are aware, there has not been much exploration of tuning *and* inference in the fashion of Eq. (13). We provide a first, formal comparison of these approaches next.

Qualitative Pros & Cons We view the approach of Eq. (13) to be preferable for a few reasons.

1. **Compatibility with Other Tasks:** Most other language modeling tasks require retention of the full token distribution. While we do not experiment with this in our paper, uncertainty tuning of implicit forecasts, as in Eq. (13) and § 2.3, can theoretically be coupled with other tasks in a more broadly scoped fine-tuning pipeline. In contrast, use of normalized probabilities during uncertainty tuning, as in Eq.(14), would require distinct loss functions and inference protocols to be coupled with other tasks (due to differences in score normalization).

2. **Generalized Extension:** Extension beyond yes/no answers (and dialogue forecasting) is far easier when using Eq. (13). Indeed, we simply change the token in the numerator and add more data instances during training. In the alternative Eq. (14), we may require additional algorithms/compute to select candidate sets – as is done by Jiang et al. (2021). For one, these added protocols can inhibit inference due to compounding errors, and related to our first point, these added protocols also make uncertainty tuning on many different types of tasks and data prohibitive.

Theoretical and Empirical Characterization of Differences We can also consider the difference between these approaches more precisely. We begin with a theoretical result that is both conceptually informative and useful for empirical study. Primarily, we bound the absolute difference between the two types of implicit forecasts. Define $\mathcal{X} = \mathcal{T} - \{\texttt{yes}, \texttt{no}\}$, then:

$$\begin{aligned}
|\hat{P}_{\texttt{IF}} - \hat{P}_{\texttt{IFN}}| &= \hat{P}_{\texttt{IFN}} \cdot |1 - \hat{P}_{\texttt{IF}}/\hat{P}_{\texttt{IFN}}| = \hat{P}_{\texttt{IFN}} \cdot \left| 1 - \frac{1}{\hat{P}_{\texttt{IFN}} + (1 - \hat{P}_{\texttt{IFN}}) + \frac{\sum_{x \in \mathcal{X}} e^{Z_x/\tau}}{e^{Z_{\texttt{yes}}/\tau} + e^{Z_{\texttt{no}}/\tau}}} \right| \\
&= \hat{P}_{\texttt{IFN}} \cdot \left| 1 - \left(1 + \frac{\sum_{x \in \mathcal{X}} e^{Z_x/\tau}}{e^{Z_{\texttt{yes}}/\tau} + e^{Z_{\texttt{no}}/\tau}} \right)^{-1} \right| \\
&= \hat{P}_{\texttt{IFN}} \cdot \left| \frac{\sum_{x \in \mathcal{X}} e^{Z_x/\tau}}{e^{Z_{\texttt{yes}}/\tau} + e^{Z_{\texttt{no}}/\tau}} \cdot \left(1 + \frac{\sum_{x \in \mathcal{X}} e^{Z_x/\tau}}{e^{Z_{\texttt{yes}}/\tau} + e^{Z_{\texttt{no}}/\tau}} \right)^{-1} \right| \leq \frac{\sum_{x \in \mathcal{X}} e^{Z_x/\tau}}{e^{Z_{\texttt{yes}}/\tau} + e^{Z_{\texttt{no}}/\tau}} \\
&\leq \frac{\sum_{x \in \mathcal{X}} e^{Z_x/\tau}}{e^{Z_{\texttt{no}}/\tau}} = \sum_x \left(\frac{e^{Z_x}}{e^{Z_{\texttt{no}}}} \right)^{1/\tau} = \sum_x \varepsilon_x^{1/\tau}
\end{aligned} \tag{15}$$

where we define $\varepsilon_x = e^{Z_x}/e^{Z_\texttt{no}}$ as the *excess* score ratio for the token x. We estimate this value empirically for temperature $\tau = 1$ across all different datasets, splits, and setups for our fine-tuned models, finding a small average of 1.1 (on a 100pt scale).

Interpretation So, for $\tau = 1$, $\hat{P}_\texttt{IF}$ and $\hat{P}_\texttt{IFN}$ are *practically equivalent* forecasts. The *main difference is the qualitative benefits of* $\hat{P}_{IF}$ we just discussed. Because x^p decreases as a function of $p > 1$ (and $x < 1$), we also know this number will not get larger for smaller τ – so, our interpretation for $\tau < 1$ remains the same. On the other hand, our observed difference does grow with τ, which can mar our interpretation if one uses post-hoc correction techniques like τ scaling. In our experimental study, this is largely unimportant, since our experiments on fine-tuned models actually observe the implicit signal at $\tau = 1$ *before* doing our proposed (simulated) correction technique in § 2.5.

All this is to say, if one conducts correction as in § 2.5, the tuned forecasting approaches are equivalent with about 1% difference on average. Consequently, it makes sense to use Eq. (13), recouping the potential qualitative benefits discussed earlier. As for use of other correction techniques, we discuss these next.

A.2 Comparison of Post-Hoc Correction Techniques

In this section, we discuss other choices of post-hoc correction, comparing them to our proposal in § 2.5. As we are the first to study language modeling of uncertainty in a conversational forecasting domain, we focus on studies calibrating uncertainty to model correctness. Namely, we consider approaches by

1. Kadavath et al. (2022), who infer an implicit signal and conduct temperature scaling using the language model's entire predicted token distribution;[9]
2. Jiang et al. (2021), who infer implicit signals and scale temperature on only a set of candidate tokens;
3. and Tian et al. (2023), who study direct signals of model correctness and briefly suggest their own temperature scaling approach for this their experiments.

Our approach is mixed. We (a) infer implicit signals using the full token distribution like Kadavath et al. (2022), and then (b) conduct an "approximate" Platt scaling (Platt et al., 1999) on only a set of candidate tokens. Recall, the "approximation" (or simulation) in step (b) is what allows the method to be applicable to *both* implicit forecasts and direct forecasts.

As a standalone property, this unification is nice. It reduces the computational overhead to study many different methods, since we can use the same inference and post-processing code for all forecasts in our experiments. In addition, we point out our approach is **computationally efficient**. Indeed, using the approach we propose, correction is done using the probability of the `yes` token only, so we need only conduct one forward pass and save this single float per instance. In contrast, temperature scaling using the entire token distribution – for implicit forecasting, as done by Kadavath et al. (2022) – requires an *increase in forward passes at least proportional to the number of temperatures we try*, or otherwise, about 32000× more memory to avoid re-computing forward passes by remembering the token scores.

Next, we conduct some theoretical and empirical analyses comparing our correction technique to that of Jiang et al. (2021), showing it is practically equivalent. Later on, we also compare our post-processing with Kadavath et al. (2022) on implicit forecasts and Tian et al. (2023) on direct forecasts, using an empirical study. Again, we find ours to be practically equivalent (or better on average).

Comparison of Proposed Correction to that of Jiang et al. (2021) Recall our "estimated logit" post-processing technique described in Eq. (12). We have processed score:

$$\hat{Z}_\texttt{yes} = \ln\big(\hat{P}/(1-\hat{P})\big)/\tau - \beta \tag{16}$$

and the post-processed forecast $\hat{P}_\texttt{PP} = \hat{P}_\texttt{new}$, expanded as below:

$$\begin{aligned}\hat{P}_\texttt{PP} = (1 + \exp(-\hat{Z}_\texttt{yes}))^{-1} = \left(1 + e^{\beta}\Big(\frac{1-\hat{P}}{\hat{P}}\Big)^{1/\tau}\right)^{-1} = \frac{\hat{P}^{1/\tau}}{\hat{P}^{1/\tau} + e^{\beta}(1-\hat{P})^{1/\tau}} \\ = \frac{e^{Z_\texttt{yes}/\tau}}{e^{Z_\texttt{yes}/\tau} + e^{\beta}(e^{Z_\texttt{no}} + \sum_x e^{Z_\texttt{x}})^{1/\tau}}\end{aligned} \tag{17}$$

[9] Recall, this is their methodology for "P(True)". Their methodology for "P(IK)" is most similar to Jiang et al. (2021).

τ	0.25	0.5	1	1.5	1.75	2	2.5
Prob. Difference Bound	4.3	2.1	1.1	0.7	0.6	0.5	0.4

Table 4: Estimation of bound in Eq. (18) from data with $\beta = 0$. Note, we report the bound on a 100pt scale.

Post-Processing Method	% Preferred (over ours)	Magnitude (Brier score gain; 100pt)
Kadavath et al. (2022)	4.2	0.03
Tian et al. (2023)	16.7	0.03
Tian et al. (2023) + Bias Correction	20.1	0.03

Table 5: Comparison of post-processing methods to our approach. For the method of Tian et al. (2023), we also consider adding a (new) second parameter – a bias correction term similar to β – to make it more competitive. We report percent of times each method is preferred (compared to our method) based on validation data, as well as average magnitude of preference. In practice, when validation data is used to select among methods, our technique is generally preferred over existing techniques. Even when others are preferred, the degree of preference is small. Meanwhile, when preferred, our post-processing method has preference magnitude 6× larger on average (0.19).

Defining $\varepsilon = \sum_x \varepsilon_x$ we have

$$\begin{aligned} |\hat{P}_{\texttt{PP}} - \hat{P}_{\texttt{IFN}}| &= \Big|\frac{e^{Z_{\texttt{yes}}/\tau}}{e^{Z_{\texttt{yes}}/\tau} + e^{\beta}(1+\varepsilon)^{1/\tau}e^{Z_{\texttt{no}}/\tau}} - \frac{e^{Z_{\texttt{yes}}/\tau}}{e^{Z_{\texttt{yes}}/\tau} + e^{Z_{\texttt{no}}/\tau}}\Big| \\ &= e^{Z_{\texttt{yes}}/\tau} \cdot \Big|\frac{e^{Z_{\texttt{no}}/\tau} - e^{\beta}(1+\varepsilon)^{1/\tau}e^{Z_{\texttt{no}}/\tau}}{e^{2Z_{\texttt{yes}}/\tau} + e^{Z_{\texttt{yes}}/T + Z_{\texttt{no}}/\tau} + e^{\beta}(1+\varepsilon)^{1/\tau}e^{Z_{\texttt{yes}}/T+Z_{\texttt{no}}/\tau} + e^{\beta}(1+\varepsilon)^{1/\tau}e^{2Z_{\texttt{no}}/\tau}}\Big| \\ &= \Big|\frac{1 - e^{\beta}(1+\varepsilon)^{1/\tau}}{e^{Z_{\texttt{yes}}/T - Z_{\texttt{no}}/\tau} + 1 + e^{\beta}(1+\varepsilon)^{1/\tau} + e^{\beta}(1+\varepsilon)^{1/\tau}e^{Z_{\texttt{no}}/T - Z_{\texttt{yes}}/\tau}}\Big| \leq |1 - e^{\beta}(1+\varepsilon)^{1/\tau}| \end{aligned} \tag{18}$$

Since β is a free parameter, it can always be chosen to be 0 during post-processing (e.g., if we determine it is not helpful based on validation data). Thus, since any deviation due to β would be by choice to improve our forecasts, we consider estimation of this bound when $\beta = 0$. Indeed, we can easily collect statistics on $(1+\varepsilon)$ during the forward passes of our experiments, and do so, estimating this bound value with different selections of τ in Table 4. We find all differences to be practically negligible.

Comparison of Proposed Correction to that of Kadavath et al. (2022) and Tian et al. (2023) Lacking theoretical analysis, we compare our unified post-processing approach to the discussed techniques of Kadavath et al. (2022) and Tian et al. (2023), empirically. To keep all methods on equal footing, we use our fine-tuned Llama-2 7b model for implicit forecasts; i.e., since the method of Kadavath et al. (2022) only operates on implicit forecasts. We also limit study to one prompt setup (the data inferred prior) because the method of Kadavath et al. (2022) is more computationally expensive. To evaluate, in Table 5, we look at performance at minimizing Brier score on the validation set, reporting the percent of times we would have preferred the method of Tian et al. (2023) or Kadavath et al. (2022) due to a lower Brier score. We also report the average magnitude of preference; i.e., how much smaller the Brier score is. The logic here is to show the utility of each post-processing method in a *practical setting*, since we would never actually select a method that does worse on validation data in practice. Indeed, the results show that if we *did* use these other proposals, in conjunction with our own, and picked the best technique for each instance based on validation data, we would have still have used our own proposal most of the time.[10] Moreover, the actual magnitude of preference for other methods is very small, so even when another method is preferred, we would ultimately expect similar performance when transferring to a test set.

Computational Details To select hyper-parameter τ and β, we consider two cases: $\beta \neq 0$ and $\beta = 0$. For the first ($\beta \neq 0$), we fit a 2 parameter logistic model of the outcome variable (i.e., this optimizes a proper scoring function – the log score). For the latter case ($\beta = 0$), we use either traditional temperature scaling (optimizing Brier score) or a newer scaling approach called *Expectation Consistency*, as described

[10] We used only our approach for experiments to avoid extra computational overhead, and because it was best overall.

by Clarté et al. (2023). To decide which of these cases (and subsequent optimization procedures) to use, we compare Brier scores, picking the method that yields the overall lowest on validation data. To re-implement the approach of Tian et al. (2023), we optimize the functional form $\exp(\log(P)/\tau)$, or more generally $\exp(\log(P)/\tau)/\beta$ when applying bias correction, picking parameters to minimize Brier score; this preserves the authors' primary suggestion that scaling be done so the result is proportional to p^α for some α.[11] In general, we use the `scipy` optimization package or `scikit-learn` to implement the aforementioned parameter selection. When re-implementing the correction approach of Kadavath et al. (2022), it is too computationally costly to use the `scipy` optimization package, so we conduct a simple linear search for $\tau \in \{0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2\}$.

[11] We inferred no other restrictions or implementation details from their description.

A.3 An Off-Policy Policy Gradient Theorem

In this section, we show the claimed result from the main text:

$$\begin{aligned}\nabla_\theta \mathbf{E}[R] = \mathbf{E}\Big[\mathsf{s}_{\tilde{T}} \cdot \tfrac{\mu_T(\tilde{T})}{\mu_{\tilde{T}}(\tilde{T})} \cdot \nabla_\theta \log \mu_T(\tilde{T})\Big] \\ \text{where} \quad T \sim \mathtt{LM}^*_\theta \circ \Phi \circ D, \\ \tilde{T} \sim \mathtt{Ref}^* \circ \Phi \circ D, \\ \text{and } \mathsf{s}_{\tilde{T}} = -\mathsf{s}(\mathsf{p} \circ \tilde{T}, O).\end{aligned} \tag{19}$$

Let all random variables be as above and fix the mass functions μ_T and $\mu_{\tilde{T}}$. Then, we have

$$\begin{aligned}&-\nabla_\theta \mathbf{E}[R] = \nabla_\theta \mathbf{E}\Bigg[\sum_{t \in \mathcal{T}^*} \mathsf{s}(\mathsf{p} \circ t, O) \cdot \mu_T(t)\Bigg] = \mathbf{E}\Bigg[\sum_{t \in \mathcal{T}^*} \mathsf{s}(\mathsf{p} \circ t, O) \cdot \nabla_\theta\, \mu_T(t)\Bigg] \\ &= \mathbf{E}\Bigg[\sum_{t \in \mathcal{T}^*} \mathsf{s}(\mathsf{p} \circ t, O) \cdot \mu_T(t) \cdot \nabla_\theta \ln \mu_T(t)\Bigg] \\ &= \mathbf{E}\Bigg[\sum_{t \in \mathcal{T}^*} \mathsf{s}(\mathsf{p} \circ t, O) \cdot \mu_T(t) \cdot \tfrac{\mu_{\tilde{T}}(t)}{\mu_{\tilde{T}}(t)} \cdot \nabla_\theta \ln \mu_T(t)\Bigg] \\ &= \mathbf{E}\Bigg[\sum_{t \in \mathcal{T}^*} \mu_{\tilde{T}}(t)\Big(\mathsf{s}(\mathsf{p} \circ t, O) \cdot \tfrac{\mu_T(t)}{\mu_{\tilde{T}}(t)} \cdot \nabla_\theta \ln \mu_T(t)\Big)\Bigg] \\ &= \mathbf{E}\Bigg[\mathsf{s}(\mathsf{p} \circ \tilde{T}, O) \cdot \tfrac{\mu_T(\tilde{T})}{\mu_{\tilde{T}}(\tilde{T})} \cdot \nabla_\theta \ln \mu_T(\tilde{T})\Bigg].\end{aligned} \tag{20}$$

So, we have our desired result.

A.4 Off-Policy Implementation Details

Next, we'll discuss some choices for the reference policy `Ref`. The formal framework we've provided actually allows us to recover variants of some existing techniques for getting language models to output forecasts in token space.

The Quantizer Lin et al. (2022) fine-tune an LM to forecast the correctness of its answers in token space for a factual question-answering task. Primarily, they use the accuracy of different question types to assign confidence levels that the LM should predict for each type. We propose to extend this idea to more general settings via clustering. Instead of assuming pre-assigned partitions, we infer the partitions by clustering the data. The average outcome of a cluster is computed and assigned to each datum in the cluster, which defines a deterministic reference policy $\mathtt{Ref}^*$ to be used in Eq. (10):

$$\mathtt{Ref}^*_{\mathsf{C}}(X) = \mathsf{p}^\dagger\Big[|C(X)|^{-1} \sum_{N \in C(X)} O_N\Big] \tag{21}$$

where $C(X)$ is the neighborhood of X, O_N is the outcome of neighbor N, and $\mathsf{p}^\dagger : [0,1] \to \mathcal{T}^*$ is an inverse for p,[12] mapping probabilities to tokens. In experiments, C is defined by k-means clustering over the internal feature representations of $\mathtt{LM}_\theta$. These representations are the average (over time) of the last hidden layer of the model, ignoring masked inputs, and they are updated each epoch when clusters are re-assigned. In practice, we pick k by hyper-parameter tuning and run k-means *individually* for each dataset, re-aggregating the cluster assignments afterwards; our motivation for this is to prevent uninformative, imbalanced cluster assignments, which may occur if clusters correlate with dataset labels.

[12] p is not bijective in general, but we can consider a subset of $\mathcal{T}^*$ – e.g., strings like "72%" – for which $\mathsf{p}^\dagger$ does exist.

The Exploiter Given any pre-trained, fixed implicit signal forecaster, we can use it to train a direct forecaster via Eq. (10). For example, assuming we have trained a LM via supervised fine-tuning (§ 2.3) and fixed its forecasting function $\hat{P}_{\texttt{IF}}$, we define the deterministic reference policy:

$$\texttt{Ref}^*_{\texttt{S}}(X) = \mathsf{p}^\dagger\big[\hat{P}_{\texttt{IF}}\big]. \tag{22}$$

This provides a nice controlled view for the differences between implicit and direct forecasting, since the direct policy is actually learning from the implicit policy. Properties of the implicit policy that do not transfer to the direct policy will be of interest. As noted in the main text, this also represents a focus on exploitation, since the implicit forecasts $\hat{P}_{\texttt{IF}}$ were designed to maximize the log-likelihood on the training data. Maximizing the log-likelihood is equivalent to minimizing the negative log-likelihood (i.e., the log score) and it is known that the log score is a proper scoring function.

The Explorer Finally, it is interesting to consider that Eq. (10) indicates the language model $\texttt{LM}^*_\theta$ can learn from any policy, even a bad one. To explore this, we suggest a context-less binomial reference policy which simply assigns random probability estimates to the dialogues. Presumably, by Eq. (10), $\texttt{LM}^*_\theta$ can observe the rewards from these estimates and begin to make "sense" of them. For binomial parameters n (number of trials) and π (success ratio) we define the reference policy:

$$\texttt{Ref}^*_{\texttt{B}}(X) = \mathsf{p}^\dagger\big[B\big]; \quad B \sim \mathrm{Bin}(n, p). \tag{23}$$

In experiments, $n = 20$ and p is the average outcome in the training data (one for each dataset).

On Policy Alternatively, we can actually use $\texttt{LM}^*_\theta$ as its own reference: $\texttt{Ref}^*_{\texttt{PPO}} = \texttt{LM}^*_\theta$. As alluded by our notation, this makes Eq. (10) equivalent to on-policy policy gradient techniques, like Proximal Policy Optimization (Schulman et al., 2017). We leave investigation of on policy learning to future work.

	# Train/Test Matches			# Test Sets Matching Majority of Train			
Split	**Topic**	**Topic+L**	**Outcome**	**Length**	**Affective**	**Non-Affective**	**Multi-party**
easy	all	all	1/2	all	all	all	all
medium	2/3	none	none	all	2/3	none	all
hard	none	none	none	none	none	2/3	1/3

Table 6: Table cells show count of test sets that *share* properties with the train set. *easy* has the most shared properties (least imbalance), while *hard* has the least shared properties (most imbalance). We operate under the assumption that greater degrees of imbalance correlate with greater degrees of difficulty in generalization, which is consistent with the domain generalization literature (Gulrajani and Lopez-Paz, 2020). Generally, by design, *imbalance of shared properties* (between train and test) *increases* from *easy* to *hard*, creating a sliding scale of difficulty for the generalization problems we study. The heldout datasets for each split are listed in Table 7. The first three columns (**Topic**, simultaneous **Topic** + **L**ength, and **Outcome**) display how many of the test datasets in the split share the column property with at least one of the train datasets. For example, in *easy* 1 out of 2 test datasets share an **Outcome** with a train dataset in the same split. The remaining columns (**Length** and so on) show how many of the test datasets in each split share the column property with *a majority* of the training datasets in the same split. For example, in *hard* 1 out of 3 test datasets has an **Affective** outcome and the majority of the train datasets have a **Non-Affective** outcome (causing the "none" designation for this column). When comparing length and multi-party similarities, we assume it is easier to generalize from long to short data or multi- to single-party data. So, "sharing" means being *at least as long* or having *at least as many parties*.

B Additional Experimental Details

Additional experimental details are provided next. In general, anything we have missed here will be available in the code, which will be made public.

B.1 Additional Dataset Details

When available, we use default train/val/test splits from each dataset's original proposal paper. When not available, we split each datatset according to a 70/15/15% split. All numbers reported in the main text are computed on the unseen test set for each individual dataset (i.e., even for the *in domain* setting).

Sampling of Training Data For every epoch of training, we sample 750 dialogues from each of the heldin training datasets of the current training split (i.e., *easy*, *medium*, or *hard*). We pick 750 because this ensures a balanced sample across the training data (all datasets have at least 750 training dialogues). Each dialogue is then randomly truncated to $K \sim \mathcal{U}\{2 \ldots L\}$ turns where L is the original dialogue's turn length. As we generally train for 5 epochs, this means the model sees roughly 19-23K partial dialogues, with some dependency in examples across epochs (for the smaller datasets). We also tried using larger, imbalanced data samples for training. In this case, we sample 5K dialogues, or as many as are available. Accounting for the datasets that have less than 5K training examples, we estimate the model sees about $4\times$ more data overall.

Sampling of Test and Validation Data For each dataset, we use the same validation and test sets across all experiments. We sample 250 dialogues from the validation split of each dataset, and randomly truncate each (in exactly the same way). We sample 550 dialogues from the test split of each dataset, again, randomly truncating each in the same way for all experiments. Some datasets have less than 550 dialogues total in their test split, in which case we use all of the available test dialogues.

Outcome Definitions Precise outcome definitions for each dataset are as follows. Most outcomes are directly annotated in their original dataset proposals. We point this out or explain how we use existing annotations to determine the outcome:

- *Wikipedia editing (Attack)*: occurrence of personal attack is directly annotated
- *Craigslist*: buyer/seller goal prices are annotated. Best deal means sale price is closer to goal price

	Matching Train Set			# Matching Train Sets		
Split / Heldout Set	**Topic**	**Topic+L**	**Outcome**	**Length**	**Affective**	**Multi-party**
easy / wiki. (attack)	deleted	deleted	reddit	4/6 long	3/6 yes	3/6 yes
easy / item allocation	camp	camp	none	-	3/6 no	-
med / craigslist	item alloc.	none	none	-	2/5 no	-
med / wiki. (deleted)	none	none	none	4/5 long	2/5 no	3/5 yes
med / camp provisions	item alloc.	none	none	-	3/5 yes	-
hard / courtroom	none	none	none	2/5 long	3/5 no	2/5 yes
hard / charity	none	none	none	2/5 long	2/5 yes	-
hard / reddit	none	none	none	2/5 long	2/5 yes	2/5 yes

Table 7: This table "shows our work" for the assertions of imbalance in Table 6. It provides a description of similarities and dissimilarities between train and test sets when each dataset is one of those heldout in the split. For each heldout test dataset, the first three columns show similarity in topic, simultaneous topic + length, noting the *specific* train dataset that shares this property. The next three columns show the *number* of similar datasets among the training data, considering length, usefulness of affective reasoning, and presence of multi-party dialogues. Length "long" is categorized by having more than 2K characters on average. Recall, when comparing length and multi-party similarities, we assume it is easier to generalize from long to short data or multi- to single-party. So, we put dashes in for "short" or single-party data to note imbalances need not be measured.

- *Camp*: satisfaction post-negotiation is directly annotated on a 5 point scale: 2 levels of unsatisfied, 1 level of neutral, 2 levels of satisfied. We use annotations directly, a camper is happy if they indicated either satisfied level.
- *Reddit*: occurrence of personal attack is directly annotated
- *Charity*: occurrence of donation is annotated
- Item allocation: occurrence of deal is annotated
- *Wikipedia editing (Deletion)*: deletion of article is annotated
- *Courtroom*: petitioner winning is annotated

B.2 Hyper-Parameters and Prompts

Hyper-Parameters and other Training Details Generally, we train for 5 epochs with a batch size of 12 using AdamW for optimization (Loshchilov and Hutter, 2017). We use 4bit QLoRA (Dettmers et al., 2023) with LoRA rank 32. On 4 NVIDIA RTX A6000 GPUs, single model training is an overnight process, so we only conduct full hyper-parameter selection (linear search) on the *medium* split using Llama-2-chat 7B to save time. We use the best hyper-parameters for Llama-2 7B on *medium* for all other train/test splits and models. For implicit forecast tuning, we pick the learning rate from the range {1e-4, 2e-5, 1e-5}. For direct forecast tuning, we pick the clipping constant ϵ from the range {0.2, 0.5, 0.8} and the learning rate from the smaller range {1e-4, 1e-5} to save time. Clustering for the Quantizer off-policy is also selected from the range {10, 20}. Log score on the in-domain validation data is used to pick the best parameters.[13] Parameter selection is fairly consistent overall, with most tuning setups preferring the highest learning rate. ϵ was always 0.5 and the number clusters for the Quantizer off-policy was 10.

Inference Parameters As noted in our discussion, implicit forecasting uses $\tau = 1$ in Eq. (13) conducting post-hoc correction using our "estimated logit" procedure, as in Eq. (12), after the fact. For sampling, to conduct direct forecasting, we typically use the default hyper-parameters indicated by the model parameters (e.g., in the API, Huggingface generation configuration, or Github repository). For GPT-4 this means temperature and top p are both set to 1. For Llama-2 models, this means temperature and top

[13] In domain generalization, its important to avoid picking parameters using the held out domains, since this can bias results (Gulrajani and Lopez-Paz, 2020).

	wiki (attack)	**craigslist**	**camp prov.**	**reddit**	**charity**	**item alloc.**	**wiki** (deleted)	**court**
BSS	8.7	-10	0.01	0.9	-0.04	23.7	36.9	0
ACC GAIN	11.4	25.8	18.1	11.5	14.3	38.5	26	17
F1 GAIN	5.9	27	18.6	10.1	18.4	34.5	26.3	23.9

Table 8: Comparison of traditional accuracy metrics and proposed forecasting metrics for GPT-4. Inference is conducted with post-hoc scaling and data prior. Probability forecasts are turned into "hard" predictions by selecting the most likely outcome for the predicted probability. Accuracy and F1 are reported as gain over a random coin flip prediction (50/50). Notably, GPT-4 always improves over a random classifier, uncertainty-aware metrics (**BSS**) provide a more nuanced analysis of performance. In contrast, probability forecasts by GPT-4 do not always improve significantly over uncertainty baselines (the average data outcome). The dichotomy in performance outcomes shows uncertainty quantification can be a distinct, more difficult problem for language models, motivating the need to evaluate this skill in addition to "hard" prediction skills.

p are set 0.6 and 0.9, respectively. The only exceptions are the pre-trained Llama-2 7b (no fine-tuning) and the Llama-2 7b with direct forecasts tuned on 4× more data, which both use greedy sampling due to clear (unsupervised) problems with the forecasts; e.g., output scores that prohibited usual sampling due to NaN values or extreme variance in token samples (200x more than greedy). For models tuned in the Zephyr style, temperature is 0.7 and top p is 1. For direct forecasting with pre-trained models, we sample up to 256 tokens to allow room for an explanation (frequently provided by some models). Tuned models generally followed the training format, so we only needed to sample up to 8 tokens in these cases. Upon sampling, our parser extracted a probability forecast by searching for percentages, ratios, and floats in [0,1] (in that order) using regex matching. Before parsing, some logic is also used to drop any non-answer explanations the model provides, i.e., this is based on keywords specified in the system prompts.

Prompts As noted in the main text, we use a standard formula for prompting all models. We provide an example of the of prompt format used for tuning Llama-2 7B below (with model tags):

```
<s>[INST] [Segment Start]
<partial dialogue inserted>
[Segment End]
In the preceding conversation segment, a group of Wikipedia
contributors are deciding whether an article should be deleted.
On average, this type of conversation ends with the article
being deleted about 56% of the time.  What is the percent
probability that this specific conversation will end with the
article being deleted?  [/INST] </s>
```

and another example:

```
<s>[INST] [Segment Start]
<partial dialogue inserted>
[Segment End]
In the preceding conversation segment, the speakers are
defending their opinions on an issue.  On average, this type
of conversation ends with a personal attack about 50% of the
time.  What is the percent probability that this specific
conversation will end with a personal attack?  [/INST] </s>.
```

For pre-trained models, we also use a consistent system prompt to constrain output format and provide more task details. Below is the prompt used for direct forecasts:

```
You are NegotiationPredictionGPT, an expert language
model at predicting the likelihood of outcomes in human
```

```
language negotiations.  You will be given the first part
of a conversation between several different speakers with
potentially different goals.  Use the first part of the
conversation to put yourself in the mindset of the speakers
and estimate the likelihood of the requested conversation
outcome for these particular speakers.  Use the keyword
"OUTCOME" to report your predicted probability for the outcome
of interest, requested by the user.  Report the probability
as a percentage using the format "OUTCOME = percent".  For
example, "OUTCOME = 70%
```

We focus on prompts for direct forecasts in these examples, but prompts for implicit forecasts are similar, changing only the main question asked (to evoke a yes/no response).

C Examples

Here, we provide some examples of negotiations the model would see during training and testing. A **wiki. editing** example where the outcome of interest is the occurrence of a personal attack:

> *Speaker 3: Material moved from anon edit for discussion. I vaguely remember this or a similar incident from a TV news program. But I thought the kids were older. Notable? Verifiable?*
>
> *”In 2005, approximately twenty sixth grade students at Reading Fleming Middle School (now Reading Fleming Intermediate School) in Flemington, New Jersey contracted syphilis after attendeding a "rainbow party".”*
>
> *Speaker 3: Rainbow party*
>
> *Speaker 2: I don’t really see much point in reporting every case of syphilis ever reported.... -*
>
> *Speaker 1: Indeed. There is no source, it sounds rather urban-legendlike, and a rainbow party is a sure ingredient for those kind of tales. Sure enough, Googling for the string ”Flemington "rainbow party" syphilis” gives 0 hits.*
>
> *Speaker 0: Source from Hunterdon Central Regional High School in Flemington, New Jersy http://central.hcrhs.k12.nj.us/bezsylko/discuss/msgReader$281?mode=day*
>
> *I don’t think a teacher would assign that if it wasn’t true and, trust me, it is. One of my friend’s sister’s was one of the girls who contracted it. So, I’d appreciate it if you didn’t accuse it of being an "urban legend".*
>
> *Speaker 1: This is ”absolutely” not a reliable source, apart from the fact that this received NO media coverage. Please stop reinserting this. When MMWR reports this, we can talk again.*

An example from **craigslist** where the outcome of interest is whether the buyer will get the best deal:

> *Speaker 0: I am interested in this apartment! Can you tell me more about it?*
>
> *Speaker 1: This apartment is located in San Pablo and close to everything! You will have a short commute to the office, the hottest stores, and the newest restaurants! The apartment has lots of closet space, two bedrooms, large windows that really brighten up the space, and an enclosed patio on the back.*
>
> *Speaker 0: Great. I'm looking for a place in that area. Is a security deposit required?*
>
> *Speaker 1: Right now we have a special . . . $99 security deposit! But you have to take advantage of the offer today!*
>
> *Speaker 0: Would you be willing to go down to $800 for the first month's rent?*
>
> *Speaker 1: I am sorry, but the rent is $1725 . . . $800 is much too low.*
>
> *Speaker 0: What about $1,200?*

An example from the **charity** discussions where the outcome of interest is a occurence of a donation:

> *Speaker 1: Hi! Have you heard of an organization called Save the Children?*
>
> *Speaker 0: I think I have once before, in a grocery store I believe*
>
> *Speaker 1: Do you mind if i give you a little information about them?*
>
> *Speaker 0: Sure, go ahead*
>
> *Speaker 1: Just some ver basic info, Save the Children is an international non-governmental organization that promotes children's rights, provides relief and helps support children in developing countries.*
>
> *Speaker 0: Are they a non profit organization?*
>
> *Speaker 1: Yes they are! They work 100% on donated funds. There is a lack of support for children in developing countries, especially in war zones. For instance, millions of Syrian children have grown up facing the daily threat of violence. In the first two months of 2018 alone, 1,000 children were reportedly killed or injured in intensifying violence. Your donation can address such problems.*
>
> *Speaker 0: Oh wow, shocking news. Do you know how many children have been helped due to this organization?*
>
> *Speaker 1: According t0 their yearend report they were able to reach 155 million children. Over 200k of those kids were in the US.*
>
> *Speaker 0: Thats awesome! Are you apart of this organization or just support them?*
>
> *Speaker 1: I am just a supporter but I would like to ask how much do you like to donate to the charity? Your donation will be directly deducted from your task payment. You can choose any amount from $0 to all your payment*

An example from the **courtroom** discussions where the outcome of interest is whether the petitioner will have a favorable decision:

> *Speaker 2: I was reading, Your Honor, from the only place that I know of that the findings of fact of the district judge are reprinted. They're in the petition – the appendix – no, Your Honor, that's their brief. The petition for certiorari, page 45(a) –*
>
> *Speaker 6: 45 (a).*
>
> *Speaker 2: Which includes the district judge's opinion and findings of fact and conclusions on this remand hearing. Thank you.*
>
> *Speaker 5: Mr. Glasser.*
>
> *Speaker 1: Thank you, Your Honor. The Kolod-Alderisio problem in our case would exist only in relation to the Florida bugging. We agree with the government. There's no real issue on Florida, but there is a very severe issue, we say, in connection with an allegedly abortive additional bugging in Georgia. I haven't spoken of that today. We've briefed it pretty completely, and I would ask the Court to watch for that item since there was some animation here at the end about the Kolod problem which I think it is currently before the Court. Now –*
>
> *Speaker 3: Well, they didn't get – they didn't make any tapes at all or get any recordings, did they, in that second incident to which you refer?*
>
> *Speaker 1: They – the agent who ran it said he didn't get the tape, and I think one other agent who was in the car with him said they didn't get any effective audible results. But, again, we had a very hard pushing hearing in which I, for one, can wait, feeling that I was entitled to make a strong appellate point against the credibility of those agents on that issue too. And, indeed, on that issue above all, they were crawling all over that part of Georgia. They were there about to score, and they were not hesitating to bug. They were bugging all over the country. We think we can't prove that they were bugging in Europe. These fellows lived with bugs. It's incredible to me that they didn't have more than that one abortive car bug in Georgia. They must have bugged Desist's room. I'm speaking of perhaps – well, all right, I'll drop that point for now because it's been thoroughly briefed. Our whole submission is sufficiently stated in the briefs. Now, on Fuller, again, may I say something that is – have been abrupt. We think this Court should withdraw its action in Fuller on the ground that certiorari there was improvidently granted and I'd like to say why. We've covered it thoroughly in our last brief. Fuller involved a telegram, we all know that, but back of that telegram was a subpoena. The police in Fuller were not defiant or willful towards existing law. The police in Fuller went to the Alaska Communications Body, whatever it's called, got voluntary relinquishment of the telegram from that body pursuant to a federal regulation and they also got a subpoena. Now, the exact details of that whole subpoena picture, I don't know for sure of myself because I haven't seen the Fuller record but I've been guided through it in consultation in clause, consultation with one of the Fuller certiorari counsel. I have the page numbers. This is covered in our last brief. Now, if there was a subpoena in Fuller for that telegram, how can Your Honors reach the question in Fuller of a violation of 605 because the very first sentence of 605 provides for subpoenas nor, at least colorably and subject to a closer scrutiny of the record in Fuller than – which Your Honors may well wish to do because Fuller is a pretty drastic decision, and to render a drastic decision like Fuller on a record that may not stand up under scholarly criticism one of these days, I would think would be something that the Court that wish to hear about.*
>
> *Speaker 7: What did the Alaska Court held in Fuller –*
>
> *Speaker 1: The Alaska Court never –*
>
> *Speaker 7: With respect to the 605 violation?*
>
> *Speaker 1: Never touch this problem that I'm talking about now. Oh, well, they touched the 605 problem.*

Speaker 7: What did it involve with respect to the 605 violation?

Speaker 1: Yes, they touched the 605, but they didn't touch the problem of subpoena pursuant to 605.

Speaker 7: What did the Alaska Court hold with respect to the 605 violation of Fuller?

Speaker 1: They held that – let me think. Now, wait a minute.

Speaker 7: There's a dissent, but the Court held that –

Speaker 1: They – oh, they held that 605 does not apply to states that they adopted the basic Schwartz line.

Speaker 7: Yes.