Partition, Prompt, Aggregate: Statistical Self-Consistency in Language Models

Patrik Wolf [*,1,2,3,4] **Thomas Kleine Buening** [2]
Andreas Krause [2] **Celestine Mendler-Dünner** [1,2,3,4]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2]ETH Zürich, Switzerland
[3]ELLIS Institute, Tübingen, Germany
[4]Tübingen AI Center, Germany

Abstract

Large language models (LLMs) are often used to perform a form of conditional inference. In-context learning is one implicit way to do this. The prompt specifies a context and the model's output is treated as an estimate of the corresponding conditional distribution. If this interpretation holds, then LLM estimates should satisfy basic probabilistic identities. In particular, the law of total probability asserts that prior-weighted conditional distributions aggregate into population-level marginals over any valid partition of the population. In this work, we investigate to what extent LLM estimates adhere to this self-consistency principle. We use binary trees as an evaluation scaffold to recursively partition a population into increasingly fine-grained subpopulations. We then prompt LLMs with verbalized subpopulation descriptions in context, aggregate the resulting estimates back into population-level estimates, and compare them across partitions of varying granularity. Applying this protocol across problem domains and frontier models, we find widespread violations of basic consistency properties. An in-depth study of persona prompting further reveals a pattern we call the *macro fallacy*: estimates reconstructed from more fine-grained subpopulation responses are often better aligned with human reference data than direct population-level estimates. The effect persists across variations in partitioning, verbalization, and estimation task. Together, these observations suggest that models possess relevant subpopulation knowledge but do not reliably propagate it into aggregate estimates. This gap establishes statistical self-consistency as an unsaturated, reference-free criterion for evaluating LLMs.

1 Introduction

"Given that a voter lives in California, how likely are they to vote Republican in the next presidential election?" When we pose such a question to a large language model, we typically interpret its answer as an estimate of a conditional probability: the prompt identifies individuals within a subpopulation, and the model estimates the probability of an outcome within this group. The same interpretation applies across forecasting, social simulation, and other application domains, where contextual information is used to elicit predictions for particular scenarios. This view also underpins theoretical accounts of in-context learning (ICL), which explain how models can use the information supplied in context to perform conditional inference [Xie et al., 2022, Müller et al., 2023, Reuter et al., 2025].

Persona prompting offers one use case of in-context learning where the conditional-inference interpretation is particularly widespread. Here, a prompt is augmented with attributes of a person, such

*Correspondence to Patrik Wolf (✉ patrik.wolf@inf.ethz.ch).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

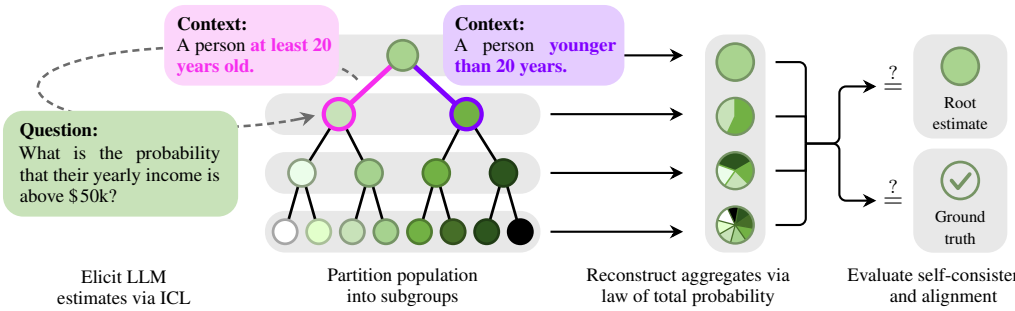


Figure 1: **Reconstructing aggregate estimates from subpopulation estimates.** A binary tree recursively partitions a base population into increasingly specific subpopulations. For each subpopulation, we elicit an LLM estimate by verbalizing the split criteria in context, then reconstruct population-level aggregate estimates at each level via the law of total probability. We compare the reconstructed aggregates against the model's direct marginal estimate to assess *self-consistency*, and against ground-truth population data to assess *alignment*.

as age, occupation, or location, and the model is asked to respond from this perspective [Argyle et al., 2023]. Outputs are then expected to approximate a conditional distribution over responses of individuals with a given attribute profile. As a testament to this perspective, distributional alignment has become a popular evaluation metric [Santurkar et al., 2023, Meister et al., 2025].

This conditional-inference interpretation implies that model outputs should obey basic laws of probability theory, yet it remains poorly understood to what extent they do. As Cruz et al. [2024] noted, if language-model predictions formed a coherent joint distribution, prompting with fewer attributes would yield the same prediction as marginalizing over the omitted attributes. We evaluate and operationalize this consistency criterion using the law of total probability. The marginal over any partition must equal the prior-weighted aggregate of its conditionals. Persona attributes serve to induce such partitions at different levels of granularity. Similarly, any pair of complementary events defines a binary partition, such as *inflation rising* versus *not rising*. Comparing the direct estimate with the aggregate of conditional estimates conditioned on elements in a partition yields a natural, reference-free self-consistency check. While self-consistency does not require any ground-truth reference data, persona prompting serves as an appealing testbed because it additionally supplies human reference statistics for alignment analyses and ablations. We use this framework to stress-test the often implicit assumption that in-context learning performs conditional inference.

Our work. We start with an in-depth study of persona prompting as an instance of in-context learning. We use *binary conditioning trees* to systematically construct personas that define subpopulations at increasing levels of specificity (see Figure 1). Let $\mathcal{X}$ denote a base population. Each successive level of the tree refines the persona description with an additional attribute. By construction, the nodes at that level define a partition $\mathcal{P}$ of the population, with $\mathcal{X} = \cup_{\mathcal{S}\in\mathcal{P}} \mathcal{S}$. Given such a partition, for any event E, the *law of total probability* [cf. Bertsekas and Tsitsiklis, 2008] asserts that

$$\underbrace{\mathbb{P}(E \,|\, X \in \mathcal{X})}_{\text{aggregate}} = \sum_{\mathcal{S}\in\mathcal{P}} \underbrace{\gamma_{\mathcal{S}}}_{\text{prior}} \underbrace{\mathbb{P}(E \,|\, X \in \mathcal{S})}_{\text{conditional}}, \tag{1}$$

where $\gamma_{\mathcal{S}} \in [0, 1]$ denotes the relative size of partition element $\mathcal{S}$. This equivalence is the core observation we build on, and we apply it in two ways. First, we substitute LLM estimates into the right-hand side of equation (1). We elicit the conditional probabilities using in-context learning and combine them using LLM-estimated prior weights. Comparing the resulting aggregate estimates for different partitions against ground-truth data reveals an interesting pattern, which we call the *macro fallacy*. Namely, the aggregate $\mathbb{P}(E \,|\, X \in \mathcal{X})$ retrieved directly from the LLM is systematically less aligned with human data than the estimate reconstructed from finer-grained conditionals and the corresponding priors via the law of total probability. The effect is robust across target quantities, models, and tree structures. Second, we introduce reference-free self-consistency checks by comparing LLM estimates of the right-hand side of (1) across partitions. Because the law of total probability requires these quantities to be equal for every partition, any disagreement reveals an internal inconsistency in the model. To implement this comparison, we propose two complementary diagnostics of statistical

self-consistency: *split consistency* and *order consistency*. Together, these checks assess statistical self-consistency across a broad class of aggregation procedures. Instantiating them in settings spanning persona prompting and synthetic forecasting across a diverse set of models, we find widespread violations even among state-of-the-art frontier models. Furthermore, consistency scores exhibit no apparent correlation with the abilities assessed by standard benchmarks.

2 Background and related work

In-context learning as conditional inference. We use *in-context learning* (ICL) in the broad sense, which encompasses general forms of learning from contextual information. This information may consist of natural language instructions, task demonstrations, or both, and includes common settings such as zero-shot prompting, few-shot prompting, persona prompting, and forecasting [Brown et al., 2020, Schulhoff et al., 2025]. A substantial body of literature interprets ICL as conditional or Bayesian inference [Xie et al., 2022, Müller et al., 2023, Reuter et al., 2025, de Carvalho et al., 2025, Buening et al., 2026]. However, there is mixed evidence on how closely LLMs satisfy this characterization [Falck et al., 2024, Chlon et al., 2025].

Persona conditioning and distributional alignment. Persona conditioning is a form of ICL in which a model is asked to respond from the perspective of a person described by sociodemographic attributes [Argyle et al., 2023, Beck et al., 2024]. Distributional alignment studies compare the resulting response distributions with those of human populations, thereby implicitly treating model outputs as conditional distributions [Santurkar et al., 2023, Meister et al., 2025, Hu et al., 2026]. This interpretation has been challenged by unreliable simulations, sensitivity to prompt formulation, degenerate output distributions, and distortions of the joint structure of human beliefs [Dominguez-Olmedo et al., 2024, Bisbee et al., 2024, Cruz et al., 2024, Lutz et al., 2025, Barrie and Cerina, 2026]. We test it directly by asking whether statistically equivalent estimators from the same model agree.

Statistical self-consistency. We define statistical self-consistency as agreement between a marginal estimate and the prior-weighted aggregate of its conditionals over any valid partition, a requirement that does not rely on reference data. This differs from the prevalent uses of the term self-consistency in the machine learning literature, where it typically refers to consistency across semantically equivalent contexts [Elazar et al., 2021, Wang et al., 2022, Li et al., 2024]. More closely related work tests probabilistic coherence under conjunctions, Bayes' rule, or logical forecasting constraints [Zhu and Griffiths, 2024, Paruchuri et al., 2024, Fluri et al., 2024, Paleka et al., 2025], as well as agreement between direct and compositionally reconstructed answers [Bottman et al., 2026]. What is unique to our work is that we compare coherence at varying levels of specificity, and against ground-truth data, which enables novel insights related to the quality of individual estimates.

3 Expressing aggregates through partitions

To instantiate the law of total probability, we systematically slice a population into recursively defined and mutually exclusive subpopulations, and then assess the consistency of LLM estimates aggregated over the partitions induced at different tree levels.

3.1 Constructing partitions using binary trees

We use a binary conditioning tree (BCT) to recursively partition a base population into increasingly fine-grained subpopulations.[2] The root node corresponds to the base population $\mathcal{X}$, and each level ℓ applies the same binary split function $\phi_\ell : \mathcal{X} \to \{0, 1\}$ to every node from the preceding level. Data points that satisfy $\phi_\ell(x) = 0$ are assigned to the left child, while data points with $\phi_\ell(x) = 1$ are assigned to the right child. As a result, the tree induces a different partition of $\mathcal{X}$ at every level.

Persona trees as a testbed. As a case study, we construct a binary tree over the population of individuals living in the United States, using data from the 2024 American Community Survey (ACS)[3] as the human reference population. The ACS data contains records for 3.8 million individuals and captures demographic, employment, health, and related attributes. Due to its large sample size,

[2] We refer to a population in a statistical sense as an arbitrary distribution over some instance space $\mathcal{X}$.

[3] https://www.census.gov/programs-surveys/acs

rigorous sampling design, and population weighting, it offers one of the best representations of the U.S. population. For a fixed set of attributes, we select balanced splits that reduce the prior-weighted within-subgroup variance of yearly income, analogous to regression trees [Breiman et al., 1984]. This yields a split by age $\phi_1(x) = \mathbb{1}\{31 \leq \text{age}(x) \leq 68\}$, followed by employment status $\phi_2(x) = \mathbb{1}\{x \text{ is employed or self-employed}\}$. In Appendix F.1, we provide additional details on split selection and data preprocessing, including the ACS attribute mappings and income binning.

3.2 Obtaining conditional LLM estimates at varying levels of granularity

For our ACS case study, we instantiate the event of interest E as the income exceeding a specified threshold, which mimics the popular UCI Adult benchmark task [Becker and Kohavi, 1996, Ding et al., 2021]. Specifically, for a threshold $\tau \in \mathbb{R}$, we take $E_\tau = \{Y > \tau\}$, where Y denotes yearly income. For any subpopulation $\mathcal{S} \subseteq \mathcal{X}$, we define the corresponding tail probability as

$$T_\tau(\mathcal{S}) \triangleq \mathbb{P}(Y > \tau \,|\, X \in \mathcal{S}).$$

We denote the LLM estimate of this quantity by $\hat{T}_\tau(\mathcal{S})$. Varying τ lets us probe binary target distributions with different degrees of entropy. The entropy is maximized when $T_\tau(\mathcal{S}) = 1/2$ and decreases as the tail probability approaches zero or one.

Verbalizing split criteria for in-context learning. To implement this conditioning structure, we pair the estimation query with a natural-language description of the subpopulation specific to each BCT node. This description consists of a base context C_0 specifying the reference population, followed by the verbalized branch restrictions along the root-to-node path. This construction follows common practice in sociodemographic prompting [Beck et al., 2024]. When the context consists only of C_0, we refer to the resulting method as *direct prompting*. The base context C_0 may also be empty, in which case the base population is left implicit. For the persona tree, we use the folktexts package [Cruz et al., 2024] to systematically translate attributes into persona descriptions. The exact prompt template is given in Appendix H.2, with prompt-sensitivity ablations reported in Appendices E.5 and E.6.

3.3 Reconstructing aggregate estimates

Aggregating conditional estimates according to the law of total probability also requires subgroup priors, which we elicit from LLMs using a level-wise protocol. For each tree level, we query the model for the prior distribution over the corresponding partition, repeat this multiple times, renormalize the raw outputs, and average the resulting estimates. We denote the resulting LLM-estimated prior of subpopulation $\mathcal{S}$ by $\hat{\gamma}_\mathcal{S}$. Further details are provided in Appendix F.2. Combining the elicited conditionals and priors yields an LLM-reconstructed aggregate at each level ℓ:

$$\hat{T}^{(\ell)}_{\tau,\text{agg}}(\mathcal{X}) \triangleq \sum\nolimits_{\mathcal{S} \in \mathcal{P}_\ell(\mathcal{X})} \hat{\gamma}_\mathcal{S}\, \hat{T}_\tau(\mathcal{S}). \tag{2}$$

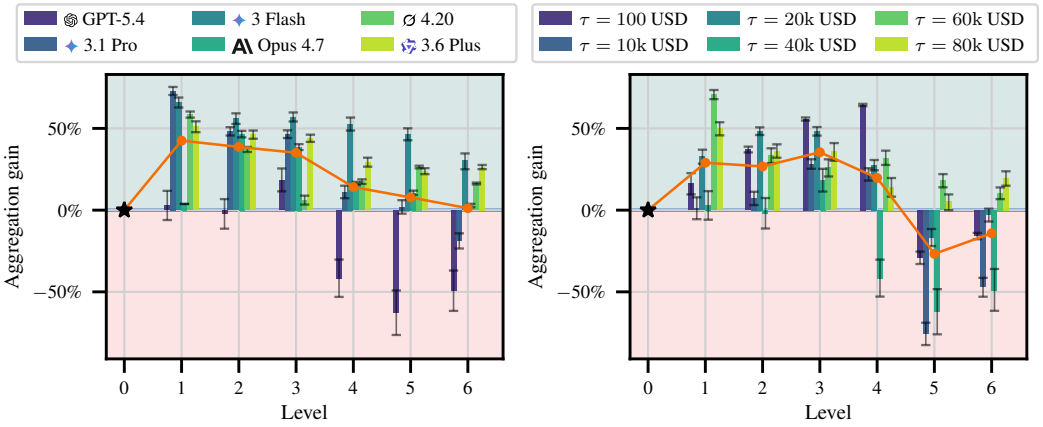
Under the conditional-inference interpretation of in-context learning, each node prompt queries the same target outcome for a specific subpopulation. The branch restrictions along the root-to-node path act as conditioning events that define this subpopulation relative to the base population. Under these assumptions, the elicited quantities $\hat{T}_\tau(\mathcal{S})$ and $\hat{\gamma}_\mathcal{S}$ should satisfy the same aggregation identity as their survey-based counterparts.

4 The macro fallacy

Each level of the persona tree induces a different partition of the base population and comes with a corresponding LLM-reconstructed aggregate estimate, computed according to (2). We start by comparing each of these estimates against survey-based reference data to understand how well they are aligned with ground-truth population statistics. More formally, for a population $\mathcal{X}$ and the partition $\mathcal{P}_\ell(\mathcal{X})$ induced by level ℓ, we define the aggregate alignment error between the reconstructed population-level estimate $\hat{T}^{(\ell)}_{\tau,\text{agg}}(\mathcal{X})$ and the ACS ground-truth tail probability $T_\tau(\mathcal{X})$ as

$$\text{AErr}(\ell) \triangleq \left| T_\tau(\mathcal{X}) - \hat{T}^{(\ell)}_{\tau,\text{agg}}(\mathcal{X}) \right|. \tag{3}$$

At level $\ell = 0$, the definition corresponds to a classical notion of alignment between the model and the data [Meister et al., 2025]. At deeper levels, the LLM estimate is instead obtained by first eliciting



(a) Threshold $\tau = 40$k USD for varying models.

(b) GPT-5.4 with varying income threshold.

Figure 2: **Relative error reduction of reconstructed aggregates over direct prompting.** We report the relative aggregation gain of the reconstructed estimate, defined as $1 - \mathrm{AErr}(\ell)/\mathrm{AErr}(0)$, as a function of tree depth ℓ, where $\mathrm{AErr}(0)$ denotes the alignment error of direct prompting. The black error bars indicate 90 % confidence intervals computed by bootstrapping with 1000 samples. The orange curve shows the average aggregation gain. Reconstructing aggregates generally improves alignment in the first few tree levels, but the gain diminishes with depth and can become negative at deeper tree levels.

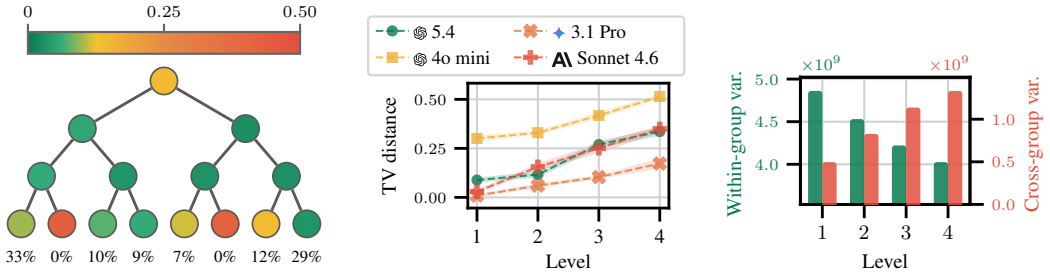
subgroup-level conditionals and then aggregating them back to the population level by using prior estimates. Thus, $\mathrm{AErr}(\ell)$ measures how well the same target quantity is recovered when the prompt is posed at the granularity induced by tree level ℓ. For estimates adhering to probability axioms, the error should not depend on ℓ. However, we find that invariance fails in practice and the alignment error varies substantially across levels.

Figure 2 shows the reduction in alignment error at deeper levels relative to direct prompting across models and problems. We refer to this reduction as the aggregation gain and use it to highlight a pattern we call the *macro fallacy*. Namely, direct root-level estimates often exhibit larger alignment errors relative to the human baseline than reconstructed estimates at lower levels. In our experiment, this pattern holds on average up to level four across models (Figure 2a) and target distributions of varying entropy (Figure 2b). We further observe that the aggregation gain profile appears concave, with a peak between levels one and four. Remarkably, this improvement arises solely from decomposing and recombining the population estimates without injecting any ground-truth statistics. Additional results, ablations, and experimental details are presented in Appendices B, E.6, F.3 and G.1.

4.1 Disentangling error sources in aggregate estimates

To better understand the findings of the previous section, we examine the quality of the conditionals and the priors separately. Figure 3a shows that conditional estimates tend to improve with depth. The improvement holds for all nodes at the first two levels, and less consistently at level three. However, the two nodes that exhibit large alignment errors have negligible prior mass and contribute little to the prior-weighted average. Conversely, priors become harder to estimate at greater depths, as reflected in the increasing total variation distance between estimated and survey-based priors across models (Figure 3b). Together, these patterns suggest that improved conditional estimates drive aggregation gains, while errors in population weighting increasingly offset the benefits of finer conditioning.

Why do conditional estimates improve with depth? One possible explanation is that partitioning reduces the burden of expressing heterogeneity in a single estimate, which is something LLMs struggle with [Bisbee et al., 2024]. Partitioning the population mitigates this issue by reducing average residual heterogeneity within subgroups while making systematic differences between them explicit. The law of total variance formalizes this decomposition as $\mathbb{V}(Y) = \mathbb{E}[\mathbb{V}(Y \mid A)] + \mathbb{V}(\mathbb{E}[Y \mid A])$, where A identifies subgroup membership. The within-group variance, $\mathbb{E}[\mathbb{V}(Y \mid A)]$, is the average variance within subgroups, while the cross-group variance, $\mathbb{V}(\mathbb{E}[Y \mid A])$, captures variation in subgroup means.



(a) Node-wise conditional alignment error. In lower levels, nodes with high alignment error tend to have small priors.

(b) Distance between LLM-estimated priors and ground-truth values. Priors get less accurate with depth.

(c) Average within-subgroup variance decreases with depth (green) while cross-subgroup variance (orange) increases.

Figure 3: **Tracing the origin of the macro fallacy.** Panels 3a and 3b separate the subgroup conditional estimates and subgroup priors in the reconstructed aggregate estimate. Node-wise conditional errors tend to improve over the root node, yet they vary substantially across the tree. Panel 3c illustrates the variance decomposition. As the population is refined, residual heterogeneity within conditioned subgroups decreases, while systematic differences between subgroups become more explicit.

As the partition is refined, variation shifts from the within-group component to the cross-group component, as illustrated in Figure 3c. Explicit decomposition thus reduces the burden of representing heterogeneous populations while directing the model's attention to individual subpopulations.

5 Statistical self-consistency

A consistent conditional estimator should satisfy the aggregation identities implied by the law of total probability over any valid partition, across trees and problem domains. In this section, we turn these criteria into two finite collections of self-consistency checks. *Split consistency* tests whether estimates elicited at two adjacent levels of a conditioning tree aggregate according to the law of total probability. *Order consistency* tests whether direct estimates are invariant under permutations of the order in which the conditioning constraints are verbalized.

General target functionals. For a feature vector $X \in \mathcal{X}$, a subpopulation $\mathcal{S} \subseteq \mathcal{X}$, and a measurable function h of the outcome Y, we define the target functional as

$$T_h(\mathcal{S}) \triangleq \mathbb{E}\big[h(Y) \,|\, X \in \mathcal{S}\big].$$

Tail probabilities are recovered as the special case $h : y \mapsto \mathbb{1}\{y > \tau\}$. Crucially, this functional obeys the aggregation identity induced by the law of total probability and is invariant to the ordering of the constraints defining $\mathcal{S}$. We use $\hat{T}_h(\mathcal{S})$ to refer to the LLM estimate elicited under a fixed canonical ordering and we use $\hat{T}_h(\mathcal{S}; \pi)$ whenever an alternative ordering π is applied to verbalize the constraints.

5.1 Split consistency

Split consistency asks whether an estimate is preserved when reconstructed over a binary attribute split. Given a subpopulation $\mathcal{S} \subseteq \mathcal{X}$ and a binary attribute A, we introduce the notation $\hat{T}_{h,\text{agg}}(\mathcal{S}; A)$ for the prior-weighted aggregate over the two subpopulations $\mathcal{S}_0 \triangleq \mathcal{S} \cap \{A = 0\}$ and $\mathcal{S}_1 \triangleq \mathcal{S} \cap \{A = 1\}$. Split consistency quantifies the discrepancy between this aggregate and the direct estimate at $\mathcal{S}$ using a metric $\text{dist}(\cdot, \cdot)$ on the space of elicited estimates.

Definition 1 (Split consistency)**.** *Let $\mathcal{S} \subseteq \mathcal{X}$ be a subpopulation and A a binary attribute. The pair $(\mathcal{S}, A)$ is ε-split consistent if*

$$\text{dist}\big(\hat{T}_h(\mathcal{S}), \hat{T}_{h,\text{agg}}(\mathcal{S}; A)\big) < \varepsilon.$$

Beyond certifying a single attribute split, we are interested in an aggregate score over a broader set of possible splits occurring in a binary tree induced by a set of multiple attributes $\mathcal{A}$. We thus collect

all pairs of conditioning events and admissible next splits induced by $\mathcal{A}$ and summarize them into a consistency score. We use $\mathcal{S}(\mathcal{J}, \boldsymbol{a}_{\mathcal{J}}) \triangleq \bigcap_{j\in\mathcal{J}}\{A_j = a_j\}$ as a shorthand for the conditioning event defined by a given realization of attributes within the index set $\mathcal{J}$.

Definition 2 (Split consistency score)**.** *Given a set of attributes $\mathcal{A} = \{A_1, \ldots, A_d\}$ and a threshold ε, the split consistency score is the fraction of satisfied checks*

$$\mathrm{SC}_\varepsilon(\mathcal{A}) \triangleq \frac{1}{|\mathcal{C}_{\mathrm{SC}}(\mathcal{A})|} \sum_{(\mathcal{S},A_k)\in\mathcal{C}_{\mathrm{SC}}(\mathcal{A})} \mathbb{1}\Big\{\mathrm{dist}\big(\hat{T}_h(\mathcal{S}), \hat{T}_{h,\mathrm{agg}}(\mathcal{S}; A_k)\big) < \varepsilon\Big\},$$

with $\mathcal{C}_{\mathrm{SC}}(\mathcal{A}) \triangleq \Big\{\big(\mathcal{S}(\mathcal{J}, \boldsymbol{a}_{\mathcal{J}}), A_k\big) : \mathcal{J} \subseteq [d],\ \boldsymbol{a}_{\mathcal{J}} \in \{0,1\}^{|\mathcal{J}|},\ k \notin \mathcal{J}\Big\}$.

The split consistency score counts the fraction of split consistency checks passed by the model. $\mathrm{SC}_\varepsilon(\mathcal{A}) = 1$ corresponds to the requirement that every admissible split consistency check is satisfied, while intermediate values quantify partial consistency. This makes the criterion useful as a benchmark score even when no model satisfies all checks.

5.2 Order consistency

Split consistency alone does not make an estimate statistically self-consistent. A complementary requirement is that the estimate assigned to a conditioning event should not depend on the order in which its defining constraints are applied. Such invariance cannot be taken for granted, as LLMs are sensitive to reordered but semantically equivalent inputs [Chen et al., 2024, Guan et al., 2025]. We formalize order invariance as a self-consistency check over a set of binary attributes.

Definition 3 (Order consistency)**.** *Let $\mathcal{S} = \mathcal{S}(\mathcal{J}, \boldsymbol{a}_{\mathcal{J}})$ be a conditioning event with $|\mathcal{J}| \geq 2$. Let $\Pi(\mathcal{J})$ denote the set of orderings of its defining constraints. We say that $\mathcal{S}$ is ε-order consistent if*

$$\mathrm{dist}\big(\hat{T}_h(\mathcal{S}; \pi), \hat{T}_h(\mathcal{S}; \pi')\big) < \varepsilon$$

for every pair of distinct orderings $\pi, \pi' \in \Pi(\mathcal{J})$.

Checking Definition 3 across all $|\mathcal{J}|!$ constraint orderings quickly becomes infeasible with increasing depth of the binary conditioning tree. We therefore restrict the evaluation to conditioning events involving exactly two attributes and construct an order consistency score defined as the fraction of satisfied checks.

Definition 4 (Order consistency score)**.** *Given an attribute set $\mathcal{A} = \{A_1, \ldots, A_d\}$, the order consistency score is the fraction of checks satisfied at tolerance ε,*

$$\mathrm{OC}_\varepsilon(\mathcal{A}) \triangleq \frac{1}{|\mathcal{C}_{\mathrm{OC}}(\mathcal{A})|} \sum_{\mathcal{S}\in\mathcal{C}_{\mathrm{OC}}(\mathcal{A})} \mathbb{1}\Big\{\mathrm{dist}\big(\hat{T}_h(\mathcal{S}; \pi_1(\mathcal{S})), \hat{T}_h(\mathcal{S}; \pi_2(\mathcal{S}))\big) < \varepsilon\Big\},$$

where $\mathcal{C}_{\mathrm{OC}}(\mathcal{A}) \triangleq \Big\{\mathcal{S}\big(\{i,j\}, (a_i, a_j)\big) : 1 \leq i < j \leq d,\ (a_i, a_j) \in \{0,1\}^2\Big\}$ *and $\pi_1(\mathcal{S})$ and $\pi_2(\mathcal{S})$ denote the two possible constraint orderings of the two-attribute event $\mathcal{S}$.*

Restricting the score to pairwise checks makes the evaluation tractable. In the next subsection, we state the condition under which these checks extend to arbitrary conditioning events.

5.3 Consistency of reconstructed aggregates

We now lift the local self-consistency checks to a uniform bound on the discrepancy between aggregates computed at different levels of arbitrary BCTs. This result needs two assumptions about the elicitation protocol and the model, which we formalize below.

Assumption 1 (Shared aggregation weights)**.** *For each partition, prior weights are elicited once using a fixed canonical ordering of the attributes and reused for every aggregation over that partition, independently of the ordering used to elicit the conditional estimates.*

Assumption 2 (Context-independent pairwise order discrepancy)**.** *For any pair of attributes A_i, A_j, any assignments a_i, a_j, and any (possibly empty) prefix, intervening, and suffix conditioning contexts P, Q, and R, the discrepancy*

$$\mathrm{dist}\big(\hat{T}_h(P, A_i = a_i, Q, A_j = a_j, R), \hat{T}_h(P, A_j = a_j, Q, A_i = a_i, R)\big)$$

depends only on the constraints $A_i = a_i$ and $A_j = a_j$, and is therefore independent of the surrounding contexts P, Q, and R.

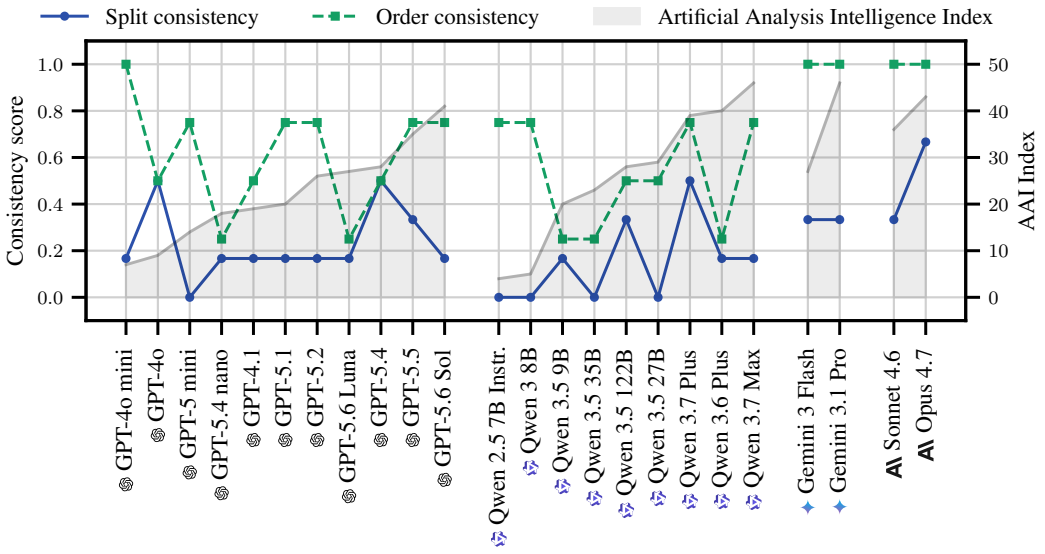


Figure 4: **Self-consistency plateaus across model generations.** We report the split consistency score $\mathrm{SC}_\varepsilon(\mathcal{A})$ and order consistency score $\mathrm{OC}_\varepsilon(\mathcal{A})$ at tolerance $\varepsilon = 0.02$ for the thresholded ACS income prediction task with $\tau = 40\text{k}$ USD. The gray shaded area shows the Artificial Analysis Intelligence (AAI) Index [Artificial Analysis, 2026] as a proxy for general model capability. Neither split nor order consistency improves systematically with increasing model capability.

Proposition 5 explains how the two proposed checks complement each other under the assumption of robust prior elicitation and context independence of pairwise order consistency. More details and the proof are given in Appendix D.3.

Proposition 5 (Informal uniform consistency bound across all trees and levels)**.** *Suppose that Assumptions 1 and 2 hold and that the distance metric is jointly convex under mixtures.*[4] *Assume that every split consistency check in $\mathcal{C}_{\mathrm{SC}}(\mathcal{A})$ is satisfied at tolerance $\varepsilon_{\mathrm{SC}}$, every pairwise order consistency check in $\mathcal{C}_{\mathrm{OC}}(\mathcal{A})$ is satisfied at tolerance $\varepsilon_{\mathrm{OC}}$, and every elicited prior at one level deviates by at most $\varepsilon_{\mathrm{PR}}$ from the prior implied by marginalizing the elicited prior at the next level. Let π and π' be any two constraint orderings that define BCTs induced by $\mathcal{A}$, both rooted at the same base population $\mathcal{X}$. If both trees have depth at most d, their corresponding level aggregates satisfy*

$$\mathrm{dist}\left(\hat{T}^{(\ell)}_{h,\mathrm{agg},\mathcal{T}}(\mathcal{X};\pi), \hat{T}^{(\ell')}_{h,\mathrm{agg},\mathcal{T}'}(\mathcal{X};\pi')\right) < 2d\,\varepsilon_{\mathrm{SC}} + 2(d-1)D_h\,\varepsilon_{\mathrm{PR}} + 2(d-1)\,\varepsilon_{\mathrm{OC}},$$

for every admissible pair of levels ℓ, ℓ'. Here, D_h is the finite diameter of the admissible estimate space, with $D_h = 1$ for normalized distance metrics.

The bound separates three sources of discrepancy: errors in split consistency, errors in prior elicitation across levels, and errors in pairwise order consistency. Under the stated assumptions, controlling these errors certifies that level aggregates remain close across different constraint orderings and levels.

6 Evaluation

Next, we demonstrate the versatility of our self-consistency checks and evaluate them across different tasks. Our self-consistency checks have the appealing property that they are reference-free, which allows us to extend our evaluation beyond surveys to domains where reliable labels are expensive, unavailable, or ill-defined. We instantiate $\mathrm{dist}(\cdot,\cdot)$ as the absolute difference for binary targets and the normalized Wasserstein-1 distance for ordered categorical targets (Appendix C.1).

Self-consistency in persona prompting. First, we evaluate self-consistency in the ACS setting discussed in prior sections and examine whether self-consistency improves with general model

[4] A metric $\mathrm{dist}(\cdot,\cdot)$ is jointly convex under mixtures if, for any weights $\lambda_1, \ldots, \lambda_n \geq 0$ satisfying $\sum_{i=1}^n \lambda_i = 1$, it holds that $\mathrm{dist}\left(\sum_i \lambda_i P_i, \sum_i \lambda_i Q_i\right) \leq \sum_i \lambda_i\ \mathrm{dist}(P_i, Q_i)$.

Model	Split consistency	Order consistency
GPT-5.4	0.33	0.50
Sonnet 4.6	0.50	0.75
Grok 4.1 Fast	0.17	0.75
Qwen 3.6 Plus	0.00	0.25

(a) Tennis forecasting

Model	Split consistency	Order consistency
GPT-5.4	0.67	0.75
Sonnet 4.6	0.33	0.75
Grok 4.1 Fast	0.17	0.00
Qwen 3.6 Plus	1.00	0.75

(b) Fantasy combat

Table 1: **Self-consistency in forecasting.** For each task, we report the split consistency score $\mathrm{SC}_\varepsilon(\mathcal{A})$ and the order consistency score $\mathrm{OC}_\varepsilon(\mathcal{A})$ at tolerance $\varepsilon = 0.02$. Higher values indicate stronger self-consistency. Self-consistency violations are widespread even in these simple synthetic settings.

capabilities assessed by standard benchmarks. We take $\mathcal{A}$ to be the demographic attributes underlying the tree in Section 3. For every conditioning event induced by $\mathcal{A}$, we elicit the model's estimated tail probability on the thresholded income task with $\tau = 40\text{k}$ USD. We then apply the self-consistency checks of Section 5 to the resulting estimates. Figure 4 shows that, despite large increases in the Artificial Analysis Intelligence (AAI) Index [Artificial Analysis, 2026] within model families, neither split nor order consistency improves systematically, with order consistency remaining generally higher. Statistical self-consistency therefore does not appear to emerge naturally with stronger overall model performance and remains far from saturated among current models. Extending this evaluation to decision models in Appendix E.3 reveals that even models trained specifically for decision tasks violate basic probabilistic consistency requirements. These findings establish statistical self-consistency as a distinct dimension of model quality, complementary to the capabilities measured by contemporary benchmarks. Implementation details and more results are deferred to Appendices E.2 and F.7.

Self-consistency in forecasting problems. Recent work on LLM forecasting, such as Prophet Arena [Yang et al., 2026], studies language models as probabilistic forecasters of real-world events and evaluates their predictions using proper scoring rules, calibration, and market-based baselines. Prior work also tests logical relations such as negation and conjunction, using consistency to diagnose failures when ground truth is unavailable [Fluri et al., 2024, Paleka et al., 2025]. We ask whether forecasts compose coherently across explicit partitions of the conditioning space. We study this requirement in two synthetic settings with binary variables encoding contextual information. In the tennis task, court surface and first-set outcome refine a prediction for a Federer–Nadal match. In the fictional fantasy-combat task, time of day and wind conditions affect the probability of a successful action. The results in Tables 1a and 1b mirror those for the ACS: self-consistency failures remain widespread across both tasks. Thus, even intuitive forecasting conditions fail to induce a coherent system of conditional forecasts. The fantasy-combat task further highlights the reference-free nature of self-consistency. Although its fictional setting provides no external ground truth against which to assess alignment, its internal consistency can still be evaluated systematically. Further details appear in Appendices F.10 and F.11.

7 Discussion

We investigate how closely the probability estimates obtained from LLMs through in-context learning conform to the law of total probability. To do so, we recursively partition populations into increasingly fine-grained subpopulations and compare two estimates of the same quantity derived at different levels of granularity. We find widespread violations of statistical self-consistency, even among frontier models. Across the models we study, statistical self-consistency shows no apparent correlation with performance on standard capability benchmarks. These findings suggest that statistical self-consistency captures an underexplored dimension of model quality on which even frontier models have room to improve.

A consequence of these violations is that statistically equivalent formulations can yield incompatible answers. This raises questions about the robustness of existing alignment claims and has implications for the design of benchmarks that assess how well LLMs estimate aleatoric uncertainty. In particular,

conclusions drawn at the aggregate level may not apply at a finer granularity. Thus, an important open question is how to reliably evaluate LLMs in light of these inconsistencies. A natural next step is to test whether optimizing for statistical self-consistency can narrow the inconsistency gap we observe. A perfectly accurate model necessarily satisfies these consistency criteria, yet satisfying them alone does not guarantee accuracy. Violations could therefore provide a flexible, reference-free training signal that complements conventional measures of accuracy.

Limitations. While our self-consistency evaluation spans multiple settings within both persona prompting and synthetic forecasting, it is necessarily incomplete, given the wide applicability of the criterion across problem domains. We also want to emphasize that self-consistency is a necessary but not sufficient condition for faithful conditional inference. The availability of ACS reference data enables a separate analysis of distributional alignment, which reveals the macro fallacy. Whether these alignment gains generalize to other domains is a promising direction for future work. Alignment measurements also depend on design choices, including verbalized probability elicitation, post-hoc normalization, and demographic splits. Our synthetic forecasting examples demonstrate the applicability of consistency checks beyond survey-based settings, but do not by themselves establish accuracy in real-world forecasting or decision-making. It remains to evaluate self-consistency across broader domains and richer conditioning structures.

Acknowledgments

We would like to thank Ana-Andreea Stoica, Vivian Nastl, Florian Dorner, and Léo le Douarec for their valuable feedback on the manuscript. We also thank the anonymous reviewers for their constructive comments on earlier versions of the paper. PW was supported by the Max Planck ETH Center for Learning Systems. PW also gratefully acknowledges G-Research for providing financial support to attend NeurIPS in Sydney. TKB was supported by an ETH AI Center Postdoctoral Fellowship. CMD acknowledges the financial support of the Hector Foundation.

Reproducibility statement

To support reproducibility, Appendix F describes the experimental setup, data preprocessing, and evaluation procedures, while Appendix H provides the exact prompt templates. The source code is publicly available at https://github.com/patrikwolf/statistical_self_consistency. Appendix D states the assumptions underlying our theoretical results and provides their proofs.

References

Lisa P. Argyle, Ethan C. Busby, Nancy Fulda, Joshua R. Gubler, Christopher Rytting, and David Wingate. Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351, February 2023. ISSN 1476-4989. doi: 10.1017/pan.2023.2.

Artificial Analysis. Artificial analysis intelligence benchmarking methodology. https://artificialanalysis.ai/methodology/intelligence-benchmarking, 2026. Accessed: 2026-07-16.

Christopher Barrie and Roberto Cerina. Synthetic personas distort the structure of human belief systems, February 2026.

Tilman Beck, Hendrik Schuff, Anne Lauscher, and Iryna Gurevych. Sensitivity, performance, robustness: Deconstructing the effect of sociodemographic prompting. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, March 2024.

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996.

Dimitri Bertsekas and John N. Tsitsiklis. *Introduction to probability*, volume 1. Athena Scientific, 2008.

James Bisbee, Joshua D. Clinton, Cassy Dorff, Brenton Kenkel, and Jennifer M. Larson. Synthetic replacements for human survey data? The perils of large language models. *Political Analysis*, 32 (4):401–416, 2024. doi: 10.1017/pan.2024.5.

Nathaniel Bottman, Yinhong Liu, and Kyle Richardson. Operadic consistency: a label-free signal for compositional reasoning failures in LLMs, 2026. URL https://arxiv.org/abs/2606.13649.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and regression trees*. Chapman and Hall/CRC, 1 edition, 1984. doi: 10.1201/9781315139470.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

Thomas Kleine Buening, Jonas Hübotter, Barna Pásztor, Idan Shenfeld, Giorgia Ramponi, and Andreas Krause. Aligning language models from user interactions. In *Third Conference on Language Modeling*, 2026.

Xinyun Chen, Ryan A. Chi, Xuezhi Wang, and Denny Zhou. Premise order matters in reasoning with large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Leon Chlon, Sarah Rashidi, Zein Khamis, and MarcAntonio M. Awada. LLMs are Bayesian, in expectation, not in realization, 2025. URL https://arxiv.org/abs/2507.11768.

André F. Cruz, Moritz Hardt, and Celestine Mendler-Dünner. Evaluating language models as risk scores. In *The Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

Gustavo Sutter Pessurno de Carvalho, Mohammed Abdulrahman, Hao Wang, Sriram Ganapathi Subramanian, Marc St-Aubin, Sharon O'Sullivan, Lawrence Wan, Luis Ricardez-Sandoval, Pascal Poupart, and Agustinus Kristiadi. Simplifying Bayesian optimization via in-context direct optimum sampling, 2025. URL https://arxiv.org/abs/2505.23913.

P. Diaconis. *Group Representations in Probability and Statistics*. IMS Lecture Notes. Institute of Mathematical Statistics, 1988. ISBN 9780940600140.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34:6478–6490, 2021.

Ricardo Dominguez-Olmedo, Moritz Hardt, and Celestine Mendler-Dünner. Questioning the survey responses of large language models. *Advances in Neural Information Processing Systems*, 37: 45850–45878, 2024.

Esin Durmus, Karina Nguyen, Thomas Liao, Nicholas Schiefer, Amanda Askell, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, Liane Lovitt, Sam McCandlish, Orowa Sikder, Alex Tamkin, Janel Thamkul, Jared Kaplan, Jack Clark, and Deep Ganguli. Towards measuring the representation of subjective global opinions in language models. In *First Conference on Language Modeling*, 2024.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, December 2021.

Fabian Falck, Ziyu Wang, and Christopher C. Holmes. Is in-context learning in large language models Bayesian? A martingale perspective. In *Forty-first International Conference on Machine Learning*, 2024.

Lukas Fluri, Daniel Paleka, and Florian Tramèr. Evaluating superhuman models with consistency checks. In *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 194–232, 2024. doi: 10.1109/SaTML59370.2024.00017.

Bryan Guan, Tanya Roosta, Peyman Passban, and Mehdi Rezagholizadeh. The order effect: Investigating prompt sensitivity to input order in LLMs, 2025. URL https://arxiv.org/abs/2502.04134.

Tiancheng Hu, Joachim Baumann, Lorenzo Lupo, Nigel Collier, Dirk Hovy, and Paul Röttger. SimBench: Benchmarking the ability of large language models to simulate human behaviors. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xiang Lisa Li, Vaishnavi Shrivastava, Siyan Li, Tatsunori Hashimoto, and Percy Liang. Benchmarking and improving generator-validator consistency of language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Marlene Lutz, Indira Sen, Georg Ahnert, Elisa Rogers, and Markus Strohmaier. The prompt makes the person(a): A systematic evaluation of sociodemographic persona prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23212–23237, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1261.

Nicole Meister, Carlos Guestrin, and Tatsunori Hashimoto. Benchmarking distributional alignment of large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, April 2025.

Samuel Müller, Matthias Feurer, Noah Hollmann, and Frank Hutter. PFNs4BO: In-context learning for Bayesian optimization. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Daniel Paleka, Abhimanyu Pallavi Sudhir, Alejandro Alvarez, Vineeth Bhat, Adam Shen, Evan Wang, and Florian Tramèr. Consistency checks for language model forecasters. In *The Thirteenth International Conference on Learning Representations*, 2025.

Akshay Paruchuri, Jake Garrison, Shun Liao, John B Hernandez, Jacob Sunshine, Tim Althoff, Xin Liu, and Daniel McDuff. What are the odds? Language models are capable of probabilistic reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11712–11733, 2024.

Arik Reuter, Tim G. J. Rudner, Vincent Fortuin, and David Rügamer. Can transformers learn full Bayesian inference in context? In *Forty-second International Conference on Machine Learning*, 2025.

Filippo Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*. Progress in Nonlinear Differential Equations and Their Applications, 87. Springer International Publishing, Cham, 1st edition, 2015. ISBN 3-319-20828-4.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect? In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 29971–30004. PMLR, July 2023.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Pham, Gerson Kroiz, Feileen Li, Hudson Tao, Ashay Srivastava, Hevander Da Costa, Saloni Gupta, Megan L. Rogers, Inna Goncearenco, Giuseppe Sarli, Igor Galynker, Denis Peskoff, Marine Carpuat, Jules White, Shyamal Anadkat, Alexander Hoyle, and Philip Resnik. The prompt report: A systematic survey of prompt engineering techniques, 2025. URL https://arxiv.org/abs/2406.06608.

U.S. Census Bureau. American Community Survey (ACS) 1-Year Public Use Microdata Sample. https://www.census.gov/programs-surveys/acs.html, 2024. Accessed: 2026-04-01.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations*, 2022.

Qingchuan Yang, Simon Mahns, Sida Li, Anri Gu, Jibang Wu, and Haifeng Xu. LLM-as-a-prophet: Understanding predictive intelligence with prophet arena. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jian-Qiao Zhu and Thomas L Griffiths. Incoherent probability judgments in large language models. *arXiv preprint arXiv:2401.16646*, 2024.

Appendices

Contents

A Illustrative reconstruction across tree levels **15**

B Disentangling conditionals and priors **15**

B.1 Conditional alignment across tree levels 16

B.2 LLM-estimated priors 17

C Extensions of self-consistency checks **18**

C.1 Discrepancy metrics for outcome distributions 19

C.2 Wasserstein distance on ordered discrete outcomes 19

C.3 Tree operations 20

C.4 Beyond binary partitions 21

C.5 Variance contraction 21

D Uniform consistency guarantees **22**

D.1 Notation and auxiliary results 22

D.2 Within-tree and cross-tree consistency 25

D.3 Additional proof 26

E Additional experimental results **26**

E.1 Partition-then-aggregate prompting 26

E.2 Self-consistency across problem domains 27

E.3 Self-consistency in decision models 29

E.4 Level-wise self-consistency 29

E.5 Prompt sensitivity 30

E.6 Sensitivity to split verbalization 31

E.7 Robustness to reasoning effort 31

F Experiment details **32**

F.1 Binary conditioning tree 33

F.2 Prior elicitation 34

F.3 Alignment error 34

F.4 Disentangling the sources of aggregation error 35

F.5 Variance decomposition across tree levels 36

F.6 Self-consistency evaluation on the binned ACS income setting 36

F.7 Self-consistency evaluation on the thresholded ACS income setting 36

F.8 Self-consistency evaluation on the ACS commute time setting 37

F.9 Self-consistency evaluation on WVS opinion distributions 37

F.10 Self-consistency evaluation of the tennis forecasting example 37
F.11 Self-consistency evaluation of the fantasy example 38

G Additional plots **39**
G.1 Aggregation gain 39
G.2 Reconstructed aggregate distributions across tree levels 40

H Prompt templates **41**
H.1 Prior elicitation 41
H.2 Sociodemographic prompting 41
H.3 Persona prompting 42
H.4 Partition-then-aggregate prompting 42
H.5 Opinion distribution elicitation 43
H.6 Tennis forecasting task 43
H.7 Fictional fantasy combat task 44
H.8 Conditional estimates from decision models 44
H.9 Prior estimates from decision models 45

I ACS attributes **46**

J WVS questions **47**

A Illustrative reconstruction across tree levels

Figure 5 provides a schematic illustration of the evaluation procedure introduced in Section 3. Starting from a base population, we partition the population into mutually exclusive and exhaustive subgroups, elicit conditional estimates and subgroup priors from the LLM, and reconstruct the population-level estimate using the law of total probability. The figure shows simplified prompts and intermediate quantities involved in the reconstruction.

B Disentangling conditionals and priors

Deviations in the alignment error across levels in Figure 2 indicate that model estimates fail to compose consistently under the law of total probability. The concave shape of the aggregation gain suggests that the models' ability to represent the population improves with depth until it starts declining as the partition becomes more fine-grained. In this appendix, we trace where these errors come from. The reconstructed estimate $\hat{T}^{(\ell)}_{\tau,\text{agg}}(\mathcal{X})$ combines two ingredients elicited from the model: the subgroup conditionals $\hat{T}_\tau(\mathcal{S})$ and the subgroup priors $\hat{\gamma}_\mathcal{S}$ used to weight them. Since the aggregate alignment error conflates both, we examine the quality of the two components separately. The survey data makes this decomposition possible because it provides reference statistics for each subgroup induced by the binary conditioning tree. We can therefore go beyond evaluating whether reconstructed population-level estimates align with the survey-based ground truth, and directly analyze how well the LLM estimates the conditional quantities at different levels of specificity. In particular, we first study an oracle-prior regime in which subgroup estimates are aggregated using ground-truth survey priors, thereby isolating the contribution of the LLM-estimated conditional probabilities to the macro fallacy. We then compare the LLM-estimated priors themselves against the survey-based subgroup proportions to assess how errors in population weighting evolve as the partition becomes finer.

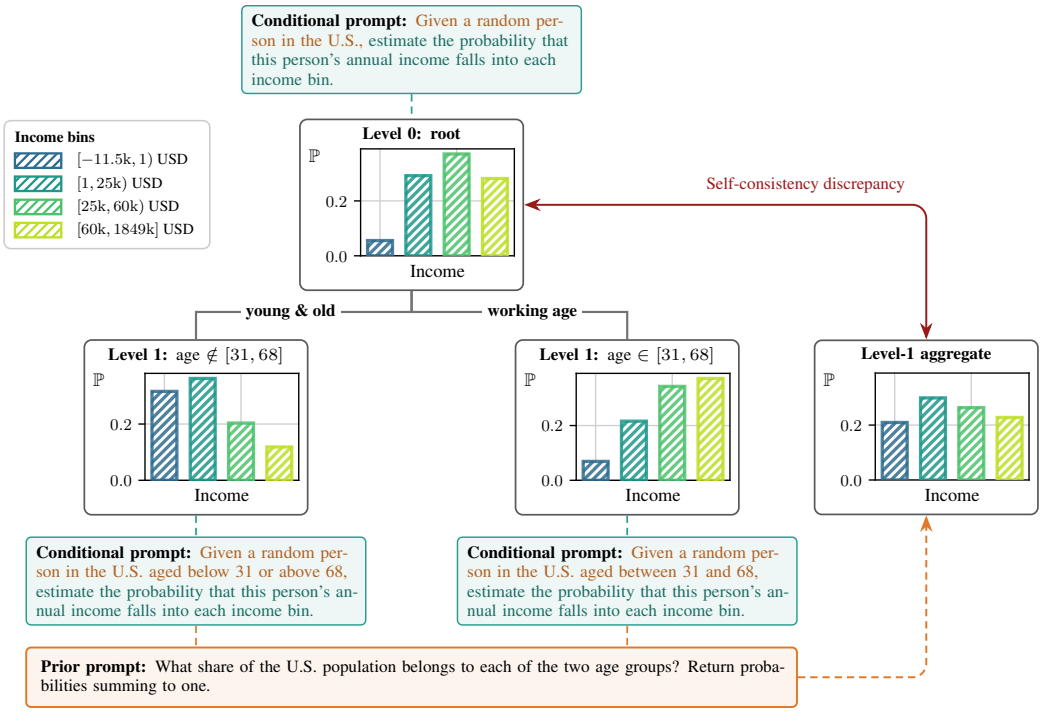


Figure 5: **Illustrative reconstruction of an aggregate estimate from subgroup estimates.** We show one refinement step of the binary conditioning tree for the ACS income-distribution task. At level 0, the LLM directly estimates the income distribution of the base population. At level 1, the population is partitioned by age into two mutually exclusive and exhaustive subgroups. The LLM estimates the conditional income distribution for each subgroup from its verbalized description. A separate prior-elicitation prompt estimates the relative sizes of the two subgroups. The resulting LLM-estimated priors are used to combine the conditional distributions into the level-1 aggregate according to the law of total probability. The normalized Wasserstein-1 distance between the direct root estimate $\hat{T}_h(\mathcal{X})$ and the reconstructed aggregate $\hat{T}_{h,\mathrm{agg}}(\mathcal{X})$ quantifies the corresponding violation of statistical self-consistency. All displayed prompts are simplified for readability.

B.1 Conditional alignment across tree levels

The aggregate alignment error in (3) is computed from both LLM-estimated subgroup conditionals $\hat{T}_\tau(\mathcal{S})$ and LLM-estimated subgroup priors $\hat{\gamma}_\mathcal{S}$, thereby conflating two sources of error. To isolate the error caused by the conditionals, we replace the LLM-estimated priors with ground-truth subgroup priors $\gamma_\mathcal{S}$ obtained from the survey data. This yields the conditional alignment error

$$\mathrm{CAErr}(\ell) \triangleq \left| T_\tau(\mathcal{X}) - \sum_{\mathcal{S}\in\mathcal{P}_\ell(\mathcal{X})} \gamma_\mathcal{S}\, \hat{T}_\tau(\mathcal{S}) \right|. \tag{4}$$

This oracle-prior regime removes uncertainty about the population composition and measures how accurately the LLM-estimated conditionals aggregate to the ground-truth population statistic.

Figure 6 reports the corresponding conditional aggregation gain, $1 - \mathrm{CAErr}(\ell)/\mathrm{CAErr}(0)$, across income thresholds. Positive values indicate that reconstructing the aggregate from subgroup estimates improves upon direct prompting. Over the first few partition depths, the reconstructed estimates are generally better aligned with the ACS reference statistic than the root-level estimate. At greater depths, however, this advantage can diminish or even reverse. Together with the node-wise results, this pattern shows that the initial aggregation gains are already present in the LLM's conditional estimates, indicating that subgroup-conditioned estimates contain useful population-level information that is not reliably elicited by direct prompting.

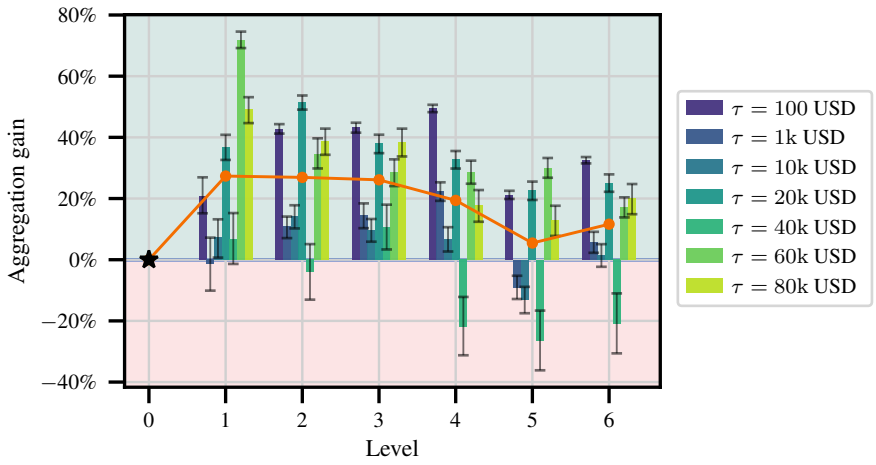


Figure 6: **Conditional aggregation gain using ground-truth priors.** On average across the considered income thresholds, reconstructing population-level estimates from increasingly fine-grained subgroup estimates using ground-truth subgroup priors improves alignment relative to direct prompting across all considered partitions. Error bars show 90 % confidence intervals computed by bootstrapping with 1000 samples.

Categorical income distributions. We next repeat the oracle-prior aggregation experiment for a categorical income target. Instead of estimating the probability of exceeding a fixed threshold, the model is asked to estimate the full distribution over income bins. To quantify alignment for such distribution-valued targets, we generalize the conditional alignment error in (4) to

$$\mathrm{CAErr}(\ell) \triangleq \overline{\mathcal{W}}_1\left(T_h(\mathcal{X}), \sum\nolimits_{\mathcal{S}\in\mathcal{P}_\ell(\mathcal{X})} \gamma_{\mathcal{S}}\, \hat{T}_h(\mathcal{S})\right), \tag{5}$$

where $\overline{\mathcal{W}}_1$ denotes the normalized Wasserstein-1 distance. This definition strictly generalizes the thresholded case. A threshold event $\{Y > \tau\}$ induces Bernoulli distributions with atoms at unit distance, for which the Wasserstein-1 distance reduces to the absolute difference of tail probabilities, recovering (4).

Figure 7 shows the reconstructed population-level income distributions at different levels of the BCT, along with the corresponding errors (5). The root-level estimate corresponds to direct elicitation of the aggregate income distribution, whereas levels one to three reconstruct this estimate from increasingly refined LLM-estimated subgroup distributions, using ACS ground-truth subgroup priors as weights. The tree structure is detailed in Appendix F.3. Consistent with the thresholded results, the reconstructed distributions are better aligned with the ACS reference distribution than the direct root-level estimate. This shows that the benefit of subgroup aggregation is not limited to binary threshold events. Further, we observe that the level-3 reconstruction is slightly less accurate than the level-2 reconstruction, suggesting that the gains from refinement are eventually offset by noisier estimates for very fine-grained subgroups.

B.2 LLM-estimated priors

Deeper partitions require estimates for a growing number of increasingly specific subgroup priors. Since aggregation gains persist to greater depths when oracle priors are used, errors in the LLM-estimated priors appear to increasingly offset the benefits of finer-grained conditioning. We investigate this possibility by comparing the estimated subgroup priors with their survey-based counterparts.

Following the level-wise elicitation protocol described in the main text and detailed in Appendix F.2, let γ_ℓ denote the survey-based distribution over the subgroups at tree level ℓ, and let $\hat{\gamma}_\ell$ denote the corresponding LLM-estimated distribution. We quantify their discrepancy using the total variation (TV) distance,

$$\mathrm{TV}(\hat{\gamma}_\ell, \gamma_\ell) = \frac{1}{2} \sum_{\mathcal{S}\in\mathcal{P}_\ell} \left|\hat{\gamma}_\ell(\mathcal{S}) - \gamma_\ell(\mathcal{S})\right|,$$

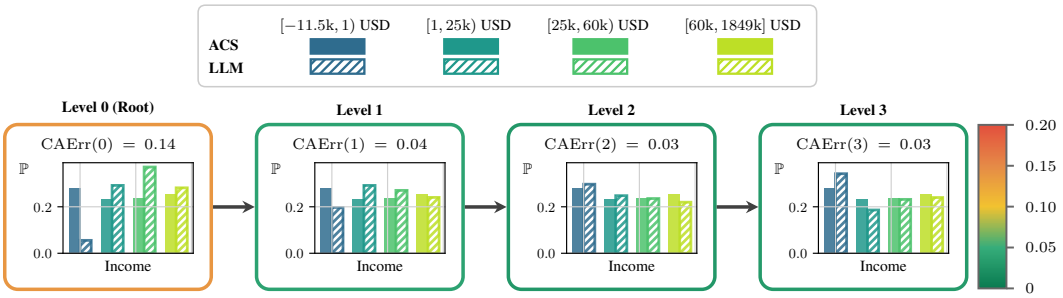
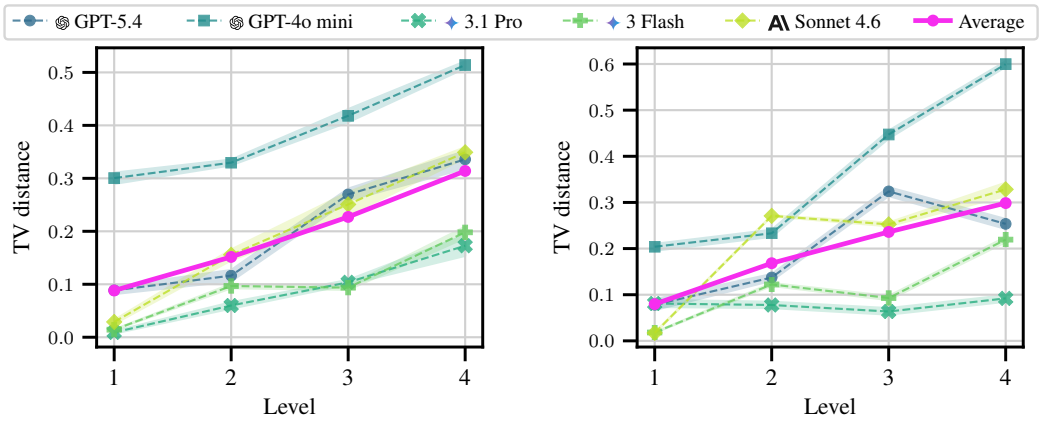


Figure 7: **Reconstructed aggregate income distributions with ground-truth priors.** We reconstruct the population-level income distribution at each tree level by aggregating LLM-estimated subgroup distributions using ACS subgroup priors. Solid bars show the survey-based aggregate distribution, while hatched bars show the aggregate reconstructed from LLM-estimated subgroup conditionals. The colored border indicates the conditional alignment error $\mathrm{CAErr}(\ell)$, with red denoting larger errors and green denoting smaller errors.



(a) BCT with splits: `AGEP`, `COW`, `WKHP`, `OCCP`

(b) BCT with splits: `COW`, `MAR`, `ESR`, `AGEP`

Figure 8: **LLM-estimated priors become less aligned at deeper tree levels.** For each tree level, we compare the LLM-estimated distribution over subgroups with the survey-based ground-truth prior distribution from the ACS data using total variation (TV) distance. Across models, prior alignment deteriorates as the tree becomes more fine-grained. Error bars show 90 % confidence intervals computed by bootstrapping with 1000 samples.

where $\mathcal{P}_\ell$ denotes the partition induced by level ℓ. Because TV distance lies in $[0, 1]$, it provides a normalized measure of prior misalignment that remains comparable across tree levels, even as the number of subgroups grows with depth.

Figure 8 shows that LLM-estimated priors become less aligned with the survey-based ground truth as the tree becomes more fine-grained. This trend is consistent across the evaluated models. While coarse population splits are estimated relatively accurately, errors increase as the model is asked to assign probability mass across more specific subpopulations. This pattern indicates that estimating the population composition of fine-grained demographic partitions remains challenging for current models and helps explain why empirical aggregation gains diminish at greater depths. Aggregation therefore entails a trade-off between improved conditional specificity and more difficult population weighting.

C Extensions of self-consistency checks

We provide additional details for the self-consistency checks introduced in Section 5. We first discuss the choice of discrepancy metric for binary and categorical distributions. We then interpret

self-consistency as operations on a binary conditioning tree. Finally, we derive the expected variance contraction criterion from the law of total variance.

C.1 Discrepancy metrics for outcome distributions

The self-consistency checks in Definitions 1 and 3 are stated for a general discrepancy metric. This allows the same aggregation principle to be applied to different types of elicited model outputs. In this subsection, we discuss how the distance metric can be defined in three canonical examples.

Binary distributions. In the ACS income setting of Sections 3 and 4, the target event is $Y > \tau$, where Y denotes yearly income in USD and τ is an income threshold. The corresponding tail probability can be written as a special case of our general target functional:

$$T_\tau(\mathcal{S}) \triangleq \mathbb{P}(Y > \tau \,|\, X \in \mathcal{S}) = \mathbb{E}\big[h(Y) \,|\, X \in \mathcal{S}\big] \qquad \text{with} \qquad h : Y \mapsto \mathbb{1}\{Y > \tau\}.$$

Thus, each estimate corresponds to the success probability of a Bernoulli distribution indicating whether an individual in subpopulation $\mathcal{S}$ has income above the threshold. For this binary target, the natural discrepancy between a direct estimate $\hat{T}_\tau(\mathcal{S})$ and the aggregated estimate $\hat{T}_{\tau,\text{agg}}(\mathcal{S})$ from (2) is the absolute difference

$$\text{dist}\big(\hat{T}_\tau(\mathcal{S}), \hat{T}_{\tau,\text{agg}}(\mathcal{S})\big) = \big|\hat{T}_\tau(\mathcal{S}) - \hat{T}_{\tau,\text{agg}}(\mathcal{S})\big|.$$

For Bernoulli distributions, the absolute difference between success probabilities coincides with the (normalized) Wasserstein-1 distance on the ordered support $\{0, 1\}$.

Ordered categorical outcomes. More generally, the outcome of interest may be categorical, with support $\{1, \ldots, K\}$. In this case, we use $\hat{T}_h(\mathcal{S})$ to denote the elicited conditional distribution over the K categories conditional on membership in $\mathcal{S}$. Similarly, $\hat{T}_{h,\text{agg}}(\mathcal{S})$ denotes the corresponding prior-weighted aggregate. The consistency error can then be defined by using an appropriate distance between probability distributions. For ordinal response options, we use the normalized Wasserstein-1 distance,

$$\text{dist}\big(\hat{T}_h(\mathcal{S}), \hat{T}_{h,\text{agg}}(\mathcal{S})\big) = \overline{\mathcal{W}}_1\big(\hat{T}_h(\mathcal{S}), \hat{T}_{h,\text{agg}}(\mathcal{S})\big) \triangleq \frac{1}{K-1} \sum_{k=1}^{K-1} \big|F(k) - F_{\text{agg}}(k)\big|,$$

where F and F_{agg} denote the corresponding cumulative distribution functions of $\hat{T}_h(\mathcal{S})$ and $\hat{T}_{h,\text{agg}}(\mathcal{S})$, respectively. This metric accounts for how far probability mass must be shifted along the response order. The normalization maps the distance to $[0, 1]$, making discrepancies comparable across questions with different numbers of response options.

Nominal categorical outcomes. For nominal response sets, the category labels have no meaningful order. We therefore use a permutation-invariant discrepancy, such as total variation distance:

$$\text{dist}\big(\hat{T}_h(\mathcal{S}), \hat{T}_{h,\text{agg}}(\mathcal{S})\big) = \text{TV}\left(\hat{T}_h(\mathcal{S}), \hat{T}_{h,\text{agg}}(\mathcal{S})\right).$$

If both distributions are represented as probability vectors $p, p_{\text{agg}} \in \Delta^{K-1}$, this is given by

$$\text{TV}\left(\hat{T}_h(\mathcal{S}), \hat{T}_{h,\text{agg}}(\mathcal{S})\right) = \frac{1}{2} \sum_{k=1}^{K} \big|p[k] - p_{\text{agg}}[k]\big|.$$

The factor $1/2$ normalizes the distance to lie in $[0, 1]$.

C.2 Wasserstein distance on ordered discrete outcomes

This subsection recalls the closed-form expression of the Wasserstein-1 distance on ordered discrete supports and explains the normalization used to compare distributions with different numbers of categories.

Lemma 6. *Let μ and ν be discrete probability measures on the ordered support $\{1, \ldots, K\}$, equipped with the metric $d(i, j) = |i - j|$, and let F_μ and F_ν denote their cumulative distribution functions. Then*

$$\mathcal{W}_1(\mu, \nu) = \sum_{k=1}^{K-1} \big|F_\mu(k) - F_\nu(k)\big|. \tag{6}$$

Proof. For probability measures on the real line, the Wasserstein-1 distance admits the classical closed form

$$\mathcal{W}_1(\mu,\nu) = \int_{\mathbb{R}} \big|F_\mu(t) - F_\nu(t)\big| \, \mathrm{d}t, \tag{7}$$

see [Santambrogio, 2015, Proposition 2.17]. Since μ and ν are supported on $\{1,\dots,K\}$, the integrand in equation (7) vanishes outside $[1,K)$ and is constant on each interval $[k,k+1)$ for $k \in [K-1]$, where it equals $|F_\mu(k) - F_\nu(k)|$. Integrating over the intervals $[k,k+1)$ for all $k \in [K-1]$ yields the result in equation (6). □

Normalization. The Wasserstein-1 distance on an ordered support, as introduced in Lemma 6, depends on the size of the support. In particular, for probability measures μ,ν supported on $\{1,\dots,K\}$ with $K \geq 2$ and metric $d(i,j) = |i-j|$, the maximum possible distance is $K-1$, attained when all mass is placed on opposite endpoints. Hence, raw Wasserstein distances are not directly comparable across multiple-choice questions with different numbers of answer options. We therefore use the normalized Wasserstein-1 distance

$$\overline{\mathcal{W}}_1(\mu,\nu) \triangleq \frac{1}{K-1}\,\mathcal{W}_1(\mu,\nu) = \frac{1}{K-1}\sum_{k=1}^{K-1}\big|F_\mu(k) - F_\nu(k)\big|.$$

This normalization maps the distance to $[0,1]$ independently of the number of ordered categories, with 0 indicating identical distributions and 1 indicating maximal separation on the support.

Bernoulli distributions. For Bernoulli distributions $\mu = \mathrm{Ber}(p)$ and $\nu = \mathrm{Ber}(q)$ on the ordered support $\{0,1\}$, equation (6) contains a single cumulative difference and therefore yields

$$\mathcal{W}_1\big(\,\mathrm{Ber}(p), \mathrm{Ber}(q)\big) = \big|(1-p) - (1-q)\big| = |p-q|.$$

Thus, the absolute error used in our binary experiments is exactly the Wasserstein-1 distance between the corresponding Bernoulli distributions.

C.3 Tree operations

The self-consistency checks introduced in Section 5 admit a natural interpretation in terms of operations on binary conditioning trees. Split consistency concerns the aggregation relations between parent and child nodes, whereas order consistency compares equivalent conditioning events obtained under different orders of the tree levels. Interchanging two adjacent levels illustrates how these two requirements complement one another.

Split consistency. Consider a node representing a conditioning event $\mathcal{S}$ and an admissible splitting attribute A_k. Refining $\mathcal{S}$ according to the two values of A_k yields the child events

$$\mathcal{S}_{k,0} = \mathcal{S} \cap \{A_k = 0\} \qquad \text{and} \qquad \mathcal{S}_{k,1} = \mathcal{S} \cap \{A_k = 1\},$$

which form a partition of $\mathcal{S}$. Split consistency requires the estimate elicited directly at the parent node to agree, up to the prescribed tolerance, with the prior-weighted aggregate of the estimates at its children. In Figure 9, this corresponds to reconstructing the orange parent node from the two red child nodes. Thus, every parent–child relation in a conditioning tree induces a local split consistency check.

A fixed conditioning tree exposes only those refinements associated with its prescribed order of split attributes. Other attributes may nevertheless define equally valid next splits of the same conditioning event. For example, after interchanging two adjacent levels in Figure 9, the pink node in the tree on the right is obtained by aggregating the two blue nodes shown in the tree on the left. Because these blue nodes belong to different subtrees of the original tree, this parent–child relation is not exposed there. This observation motivates defining the split consistency score in Definition 2 over all pairs of conditioning events and admissible next splits, rather than only over the parent–child relations of a single fixed tree.

Order consistency. Interchanging tree levels also changes the order in which the corresponding attribute constraints are presented to the model. For example, the tree on the left of Figure 9 first splits by sex and then by age, whereas the tree on the right first splits by age and then by sex. Nevertheless, corresponding leaves define the same conditioning events. Order consistency therefore requires the direct estimates elicited for corresponding leaves to agree, up to the prescribed tolerance.

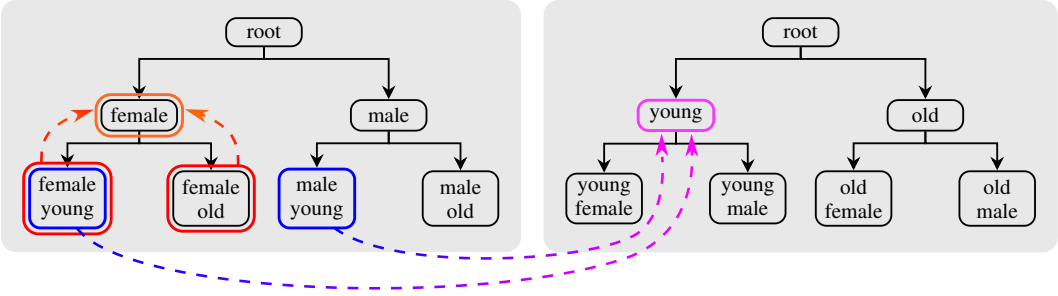


Figure 9: **Tree interpretation of split and order consistency.** Within each tree, a parent estimate should agree with the prior-weighted aggregate of its two child estimates. Interchanging two adjacent levels preserves the leaf events but changes both the order in which their constraints are presented and the intermediate conditioning events exposed by the tree.

C.4 Beyond binary partitions

Throughout the paper, we construct partitions using binary conditioning trees. This is a design choice rather than a conceptual restriction. The law of total probability in (1) holds for any valid partition of the population. In particular, suppose a subpopulation $\mathcal{S}$ is partitioned into K mutually disjoint and jointly exhaustive child subpopulations $\mathcal{S}_1, \ldots, \mathcal{S}_K$. The corresponding aggregate estimate is

$$\hat{T}_{h,\text{agg}}(\mathcal{S}) = \sum_{k=1}^{K} \hat{\gamma}_k \hat{T}_h(\mathcal{S}_k),$$

where $\hat{\gamma}_k$ denotes the estimated conditional prevalence of $\mathcal{S}_k$ within $\mathcal{S}$. The split consistency check in Definition 1 extends directly by comparing this K-term aggregate with the direct estimate $\hat{T}_h(\mathcal{S})$.

We nevertheless adopt binary trees for two practical reasons. First, binary splits do not restrict the granularity of the analysis. Any K-way partition can be represented by a sequence of $K-1$ recursive binary splits, so binary trees generate arbitrarily fine partitions of the base population. Second, binary splits simplify the construction of valid partitions. Defining one child as the complement of the other within the parent guarantees by construction that every split is mutually exclusive and jointly exhaustive, as required by the law of total probability. This makes it straightforward to build valid conditioning trees at varying levels of granularity.

C.5 Variance contraction

The self-consistency checks in Section 5 constrain elicited estimates through first-moment identities. The law of total probability prescribes exactly how conditional estimates must aggregate across a partition. The same reasoning extends to higher-order moments of the outcome distribution. In this subsection, we illustrate this with an auxiliary check derived from the law of total variance. In particular, we reduce the law of total variance to a one-sided constraint. Refining a subpopulation can only reduce the outcome variance in expectation, since the variance of the conditional means across the children is non-negative.

Definition 7 (Expected variance contraction)**.** *Given a tree $\mathcal{T}$, we say a model satisfies expected variance contraction if for every internal node v it holds that*

$$\hat{\mathbb{V}}(\mathcal{S}_v) \geq \hat{\gamma}_L \hat{\mathbb{V}}(\mathcal{S}_{v,L}) + \hat{\gamma}_R \hat{\mathbb{V}}(\mathcal{S}_{v,R}),$$

where $\hat{\mathbb{V}}(\cdot)$ denotes the variance implied by the model's estimated (conditional) distribution.

We now derive the expected variance contraction condition used in Definition 7 from the law of total variance. For two random variables Y and A, the law of total variance states that

$$\mathbb{V}(Y) = \mathbb{E}\big[\mathbb{V}(Y \mid A)\big] + \mathbb{V}\big(\mathbb{E}[Y \mid A]\big).$$

Since the second term is non-negative, conditioning on A can only reduce variance in expectation:

$$\mathbb{V}(Y) \geq \mathbb{E}\big[\mathbb{V}(Y \mid A)\big]. \tag{8}$$

Importantly, this is an averaged statement over the values of A; it does not imply that the conditional variance decreases for every realization of A individually. We now instantiate (8) at an internal node of a binary conditioning tree. Let $\mathcal{S}$ be a parent subpopulation with children $\mathcal{S}_L$ and $\mathcal{S}_R$, and let A encode child membership within $\mathcal{S}$. That is, $A = L$ corresponds to $X \in \mathcal{S}_L$ and $A = R$ corresponds to $X \in \mathcal{S}_R$, conditional on $X \in \mathcal{S}$. Applying (8) within the parent subpopulation $\mathcal{S}$ yields

$$\mathbb{V}(Y \,|\, X \in \mathcal{S}) \geq \gamma_L \, \mathbb{V}(Y \,|\, X \in \mathcal{S}_L) + \gamma_R \, \mathbb{V}(Y \,|\, X \in \mathcal{S}_R),$$

where $\gamma_L = \mathbb{P}(X \in \mathcal{S}_L \,|\, X \in \mathcal{S})$ and $\gamma_R = \mathbb{P}(X \in \mathcal{S}_R \,|\, X \in \mathcal{S})$ are the child priors within the parent. Thus, if the LLM estimates form a self-consistent system of conditional distributions, the same inequality should hold for the corresponding LLM-estimated variances, denoted by $\hat{\mathbb{V}}$. In our check, we replace the population variances and child priors by the corresponding model-induced quantities, yielding the condition stated in Definition 7.

D Uniform consistency guarantees

This appendix develops the technical results underlying our uniform consistency guarantee. We first establish auxiliary bounds and then combine them to control aggregate estimates across tree levels and tree structures.

D.1 Notation and auxiliary results

Throughout this appendix, let $\mathcal{T}$ be a binary conditioning tree of depth $L \leq d$, induced by $\mathcal{A} = \{A_1, \ldots, A_d\}$ and rooted at the base population $\mathcal{X}$. We index constraints by their attribute, so that constraint i is the constraint imposed on A_i, and we index positions by $\{1, \ldots, d\}$ likewise. A *constraint ordering* is then a bijection $\pi : \{1, \ldots, d\} \to \{1, \ldots, d\}$, where $\pi(i)$ is the position assigned to constraint i. The canonical ordering is denoted by $\overline{\pi}$. For a conditioning event $\mathcal{S}$, we write $\pi_{\mathcal{S}}$ and $\overline{\pi}_{\mathcal{S}}$ for the restrictions of these orderings to the constraints defining $\mathcal{S}$. For $r \in \{0, \ldots, L\}$, let $\mathcal{L}_r(\mathcal{T})$ denote the nodes at level r and let $\hat{\gamma}_r$ be the LLM-estimated prior distribution on $\mathcal{L}_r(\mathcal{T})$. For $v \in \mathcal{L}_r(\mathcal{T})$, denote its conditioning event by $\mathcal{S}_v$ and write $\pi_v \triangleq \pi_{\mathcal{S}_v}$. We define the aggregate estimate obtained from level r as

$$\hat{T}^{(r)}_{h,\mathrm{agg}}(\mathcal{X}; \pi) \triangleq \sum_{v \in \mathcal{L}_r(\mathcal{T})} \hat{\gamma}_r(v) \, \hat{T}_h(\mathcal{S}_v; \pi_v),$$

which at $r = 0$ reduces to the direct root estimate $\hat{T}_h(\mathcal{X})$, independently of π. To compare aggregates at adjacent levels, we marginalize the child-level prior onto the parent-level partition. Specifically, since every level-$(r+1)$ node has a unique parent at level r, the prior $\hat{\gamma}_{r+1}$ induces a prior over $\mathcal{L}_r(\mathcal{T})$ by summing the masses of each parent's children:

$$\hat{\gamma}^{\uparrow}_{r+1}(v) \triangleq \sum_{b \in \{0,1\}} \hat{\gamma}_{r+1}(v, b).$$

Because the priors are elicited separately at each level, $\hat{\gamma}_r$ need not agree with the marginal of $\hat{\gamma}_{r+1}$. We quantify their deviation by

$$\delta_r(\mathcal{T}) \triangleq \mathrm{TV}(\hat{\gamma}_r, \hat{\gamma}^{\uparrow}_{r+1}).$$

To define the local relative size weights used in the split consistency check, fix a level $r \in \{0, \ldots, L-1\}$ and let $A_{k(r)}$ be the attribute on which every node at that level is split. For $v \in \mathcal{L}_r(\mathcal{T})$, write $\{(v, b)\}_{b \in \{0,1\}}$ for its two children. If $\hat{\gamma}^{\uparrow}_{r+1}(v) > 0$, the local weights are obtained by restricting the level-$(r+1)$ prior to the children of v and renormalizing:

$$\hat{\gamma}_{k(r),b \,|\, \mathcal{S}_v} \triangleq \frac{\hat{\gamma}_{r+1}(v, b)}{\hat{\gamma}^{\uparrow}_{r+1}(v)}, \qquad b \in \{0, 1\}. \tag{9}$$

For an ordering π, define the local split aggregate at $v \in \mathcal{L}_r(\mathcal{T})$ by

$$\hat{T}_{h,\mathrm{agg}}(\mathcal{S}_v; A_{k(r)}; \pi) \triangleq \sum_{b \in \{0,1\}} \hat{\gamma}_{k(r),b \,|\, \mathcal{S}_v} \, \hat{T}_h\big(\mathcal{S}_{v,b}; \pi_{\mathcal{S}_{v,b}}\big).$$

If $\hat{\gamma}^{\uparrow}_{r+1}(v) = 0$, we use the convention $\hat{T}_{h,\mathrm{agg}}(\mathcal{S}_v; A_{k(r)}; \pi) = \hat{T}_h(\mathcal{S}_v; \pi_v)$.[5] Whenever no ordering is specified, the canonical ordering is used. For two orderings π and π' of the same set of m constraints, let $\mathrm{cay}(\pi, \pi')$ denote the transposition distance, defined as the minimum number of pairwise swaps required to transform π into π'. We define $\sigma \triangleq \pi' \circ \pi^{-1}$ as the permutation of positions that carries the arrangement π to the arrangement π'. Letting $c(\sigma)$ be its number of cycles (counting fixed points), the transposition distance satisfies Cayley's identity $\mathrm{cay}(\pi, \pi') = m - c(\sigma)$ [Diaconis, 1988, pp. 112, 118]. Since $c(\sigma) \geq 1$, this yields the diameter bound

$$\mathrm{cay}(\pi, \pi') \leq m - 1, \tag{10}$$

with equality precisely when σ is an m-cycle. Because a BCT splits on one shared attribute per level, every conditioning event $\mathcal{S}_v$ with $v \in \mathcal{L}_r(\mathcal{T})$ is defined by the same set of r constraints. Thus, $\mathrm{cay}(\pi_v, \overline{\pi}_v)$ does not depend on v. We denote this common value by $\kappa_r(\mathcal{T}, \pi)$, with $\kappa_0(\mathcal{T}, \pi) \triangleq 0$. Applying (10) with $m = r$ then gives

$$\kappa_r(\mathcal{T}, \pi) \leq \max\{r - 1,\, 0\}. \tag{11}$$

Further, we assume that the discrepancy metric $\mathrm{dist}(\cdot, \cdot)$ is jointly convex under finite mixtures. That is, for any weights $\lambda_1, \ldots, \lambda_n \geq 0$ satisfying $\sum_{i=1}^n \lambda_i = 1$, it holds that

$$\mathrm{dist}\left(\sum\nolimits_i \lambda_i P_i, \sum\nolimits_i \lambda_i Q_i\right) \leq \sum_i \lambda_i \,\mathrm{dist}(P_i, Q_i).$$

Finally, let $\mathcal{E}_h$ be the convex hull of the admissible values of the target functional T_h, and define its diameter under the discrepancy metric as

$$D_h \triangleq \sup_{U,V \in \mathcal{E}_h} \mathrm{dist}(U, V).$$

We assume that D_h is finite. For the normalized discrepancy metrics and target spaces used in the paper, we have $D_h = 1$.

Lemma 8 (Mixture sensitivity to weights)**.** *For estimates $T_1, \ldots, T_n \in \mathcal{E}_h$ and probability vectors $\alpha, \beta \in \Delta^{n-1}$,*

$$\mathrm{dist}\left(\sum_{i=1}^n \alpha_i T_i, \sum_{i=1}^n \beta_i T_i\right) \leq D_h \,\mathrm{TV}(\alpha, \beta).$$

Proof. Let $\tau = \mathrm{TV}(\alpha, \beta)$ and $c_i = \min\{\alpha_i, \beta_i\}$. The two mixtures share mass $\sum_i c_i = 1 - \tau$. If $0 < \tau < 1$, we decompose both mixtures into their shared part with weight $1 - \tau$ and their respective residual parts with weight τ:

$$T_{\mathrm{sh}} \triangleq \frac{1}{1-\tau} \sum_{i=1}^n c_i T_i, \qquad T_\alpha^{\mathrm{res}} \triangleq \frac{1}{\tau} \sum_{i=1}^n (\alpha_i - c_i) T_i, \qquad T_\beta^{\mathrm{res}} \triangleq \frac{1}{\tau} \sum_{i=1}^n (\beta_i - c_i) T_i.$$

Joint convexity then gives

$$\mathrm{dist}\left(\sum_{i=1}^n \alpha_i T_i, \sum_{i=1}^n \beta_i T_i\right) \leq (1-\tau) \underbrace{\mathrm{dist}(T_{\mathrm{sh}}, T_{\mathrm{sh}})}_{=\,0} + \tau \,\mathrm{dist}(T_\alpha^{\mathrm{res}}, T_\beta^{\mathrm{res}}) \leq \tau D_h.$$

The cases $\tau = 0$ and $\tau = 1$ follow directly. □

Lemma 9 (Split consistency across levels)**.** *Fix $0 \leq \ell < \ell' \leq L$. If every internal node at levels $\ell, \ldots, \ell' - 1$, paired with its split attribute, is $\varepsilon_{\mathrm{SC}}$-split consistent under the canonical ordering, then*

$$\mathrm{dist}\left(\hat{T}^{(\ell)}_{h,\mathrm{agg}}(\mathcal{X}; \overline{\pi}), \hat{T}^{(\ell')}_{h,\mathrm{agg}}(\mathcal{X}; \overline{\pi})\right) < (\ell' - \ell)\varepsilon_{\mathrm{SC}} + D_h \sum_{j=\ell}^{\ell'-1} \delta_j(\mathcal{T}),$$

where $\delta_r(\mathcal{T})$ is the adjacent-level prior-refinement discrepancy.

[5] The choice does not affect the level-$(r+1)$ aggregate because v receives zero mass under the induced prior. Setting its split aggregate equal to the direct estimate makes the associated split consistency check trivially satisfied.

Proof. By the triangle inequality, we have

$$\operatorname{dist}\left(\hat{T}^{(\ell)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi}),\hat{T}^{(\ell')}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi})\right)\le\sum_{j=\ell}^{\ell'-1}\operatorname{dist}\left(\hat{T}^{(j)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi}),\hat{T}^{(j+1)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi})\right).$$

Thus, it suffices to bound the discrepancy between each pair of adjacent levels. For fixed $\ell\le j<\ell'$, grouping the level-$(j+1)$ nodes by their parents using the lifted relative size estimates (9) yields

$$\begin{aligned}\hat{T}^{(j+1)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi})&=\sum_{v\in\mathcal{L}_j(\mathcal{T})}\hat{\gamma}^{\uparrow}_{j+1}(v)\sum_{b\in\{0,1\}}\hat{\gamma}_{k(j),b\,|\,\mathcal{S}_v}\,\hat{T}_h\big(\mathcal{S}_{v,b};\overline{\pi}_{\mathcal{S}_{v,b}}\big)\\&=\sum_{v\in\mathcal{L}_j(\mathcal{T})}\hat{\gamma}^{\uparrow}_{j+1}(v)\,\hat{T}_{h,\mathrm{agg}}(\mathcal{S}_v;A_{k(j)};\overline{\pi}).\end{aligned}$$

Using the shorthand notation $Q_v\triangleq\hat{T}_{h,\mathrm{agg}}(\mathcal{S}_v;A_{k(j)};\overline{\pi})$ and inserting the intermediate mixture $\sum_v\hat{\gamma}_j(v)\,Q_v$ gives

$$\begin{aligned}\operatorname{dist}\left(\hat{T}^{(j)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi}),\hat{T}^{(j+1)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi})\right)&\le\operatorname{dist}\left(\hat{T}^{(j)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi}),\textstyle\sum_{v\in\mathcal{L}_j(\mathcal{T})}\hat{\gamma}_j(v)\,Q_v\right)\\&\quad+\operatorname{dist}\left(\textstyle\sum_{v\in\mathcal{L}_j(\mathcal{T})}\hat{\gamma}_j(v)\,Q_v,\sum_{v\in\mathcal{L}_j(\mathcal{T})}\hat{\gamma}^{\uparrow}_{j+1}(v)\,Q_v\right).\end{aligned}\tag{12}$$

The first summand can be bounded by using joint convexity and local split consistency:

$$\operatorname{dist}\left(\hat{T}^{(j)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi}),\textstyle\sum_{v\in\mathcal{L}_j(\mathcal{T})}\hat{\gamma}_j(v)\,Q_v\right)\le\sum_{v\in\mathcal{L}_j(\mathcal{T})}\hat{\gamma}_j(v)\,\operatorname{dist}\big(\hat{T}_h(\mathcal{S}_v;\overline{\pi}_v),Q_v\big)<\varepsilon_{\mathrm{SC}}.$$

By Lemma 8, the second summand is bounded by

$$\operatorname{dist}\left(\textstyle\sum_{v\in\mathcal{L}_j(\mathcal{T})}\hat{\gamma}_j(v)\,Q_v,\sum_{v\in\mathcal{L}_j(\mathcal{T})}\hat{\gamma}^{\uparrow}_{j+1}(v)\,Q_v\right)\le D_h\,\delta_j(\mathcal{T}).$$

Combining these bounds with (12) yields

$$\operatorname{dist}\left(\hat{T}^{(j)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi}),\hat{T}^{(j+1)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi})\right)<\varepsilon_{\mathrm{SC}}+D_h\delta_j(\mathcal{T}).$$

Summing over $j=\ell,\dots,\ell'-1$ proves the claim. □

Lemma 10 (Sufficiency of pairwise checks)**.** *Suppose Assumption 2 holds and every (pairwise) check in $\mathcal{C}_{\mathrm{OC}}(\mathcal{A})$ is satisfied at tolerance ε. Let $\mathcal{S}=\mathcal{S}(\mathcal{J},\boldsymbol{a}_{\mathcal{J}})$ be any conditioning event induced by $\mathcal{A}$. Then, for any two distinct orderings $\pi\neq\pi'$ of its defining constraints,*

$$\operatorname{dist}\big(\hat{T}_h(\mathcal{S};\pi),\hat{T}_h(\mathcal{S};\pi')\big)<\operatorname{cay}(\pi,\pi')\,\varepsilon\le(|\mathcal{J}|-1)\,\varepsilon.$$

Proof. Since $\pi\neq\pi'$, the quantity $q\triangleq\operatorname{cay}(\pi,\pi')$ satisfies $q\ge1$. Let $\sigma\triangleq\pi'\circ\pi^{-1}$. By definition of the transposition distance, σ admits a decomposition into q transpositions, $\sigma=\tau_q\cdots\tau_1$. Set $\pi^{(t)}\triangleq\tau_t\cdots\tau_1\,\pi$ for $t\in\{0,\dots,q\}$, so that $\pi^{(0)}=\pi$ and $\pi^{(q)}=\pi'$. Consecutive orderings $\pi^{(t-1)}$ and $\pi^{(t)}$ differ by exchanging the positions of exactly two constraints with all remaining constraints held in place. By Assumption 2, the discrepancy induced by this swap equals the discrepancy of the corresponding two-attribute check in $\mathcal{C}_{\mathrm{OC}}(\mathcal{A})$. Because every such check is satisfied at tolerance ε, we obtain, for every $t\in\{1,\dots,q\}$,

$$\operatorname{dist}\Big(\hat{T}_h\big(\mathcal{S};\pi^{(t-1)}\big),\hat{T}_h\big(\mathcal{S};\pi^{(t)}\big)\Big)<\varepsilon.$$

The triangle inequality therefore gives

$$\begin{aligned}\operatorname{dist}\big(\hat{T}_h(\mathcal{S};\pi),\hat{T}_h(\mathcal{S};\pi')\big)&\le\sum_{t=1}^{q}\operatorname{dist}\Big(\hat{T}_h\big(\mathcal{S};\pi^{(t-1)}\big),\hat{T}_h\big(\mathcal{S};\pi^{(t)}\big)\Big)\\&<q\,\varepsilon=\operatorname{cay}(\pi,\pi')\,\varepsilon,\end{aligned}$$

which proves the first claim. The second claim uses the fact that the transposition distance between two orderings of $|\mathcal{J}|$ constraints is at most $|\mathcal{J}|-1$. □

D.2 Within-tree and cross-tree consistency

The preceding lemma lifts local pairwise order consistency checks to arbitrary conditioning events. Together with Lemma 9, it enables uniform comparisons of aggregate estimates across tree levels. Specifically, the following theorem shows that the discrepancy between the aggregate estimates at any two levels is controlled by three quantities: the number of intervening splits, the prior-refinement discrepancy accumulated across those splits, and the deviation of the path orderings at the two endpoint levels from their canonical orderings.

Theorem 11 (Uniform consistency across tree levels)**.** *Let $\mathcal{A} = \{A_1, \ldots, A_d\}$, and suppose that*

1. *every $(\mathcal{S}, A_k) \in \mathcal{C}_{\mathrm{SC}}(\mathcal{A})$ is $\varepsilon_{\mathrm{SC}}$-split consistent under the canonical ordering of the constraints defining $\mathcal{S}$;*
2. *every pairwise check in $\mathcal{C}_{\mathrm{OC}}(\mathcal{A})$ is satisfied at tolerance $\varepsilon_{\mathrm{OC}}$; and*
3. *Assumptions 1 and 2 hold.*

Let $\mathcal{T}$ be any binary conditioning tree of depth $L \le d$, induced by $\mathcal{A}$, rooted at $\mathcal{X}$, and equipped with a constraint ordering π. Then, for every $0 \le \ell < \ell' \le L$,

$$\operatorname{dist}\left(\hat{T}^{(\ell)}_{h,\mathrm{agg}}(\mathcal{X};\pi), \hat{T}^{(\ell')}_{h,\mathrm{agg}}(\mathcal{X};\pi)\right) < (\ell' - \ell)\,\varepsilon_{\mathrm{SC}} + D_h \sum_{j=\ell}^{\ell'-1} \delta_j(\mathcal{T}) + \left[\kappa_\ell(\mathcal{T},\pi) + \kappa_{\ell'}(\mathcal{T},\pi)\right] \varepsilon_{\mathrm{OC}},$$

where $\kappa_r(\mathcal{T},\pi)$ denotes the minimum number of pairwise swaps required to transform the attribute ordering at level r induced by π into the canonical order.

Proof. We first bound the effect of replacing the path ordering by the canonical ordering at a fixed level r. Since all pairwise checks in $\mathcal{C}_{\mathrm{OC}}(\mathcal{A})$ are satisfied, Assumption 2 and Lemma 10 imply that, for every $v \in \mathcal{L}_r(\mathcal{T})$,

$$\operatorname{dist}\Big(\hat{T}_h(\mathcal{S}_v;\pi_v), \hat{T}_h(\mathcal{S}_v;\overline{\pi}_v)\Big) \le \operatorname{cay}(\pi_v, \overline{\pi}_v)\,\varepsilon_{\mathrm{OC}} \tag{13}$$

with strict inequality whenever $\pi_v \neq \overline{\pi}_v$. When $\pi_v = \overline{\pi}_v$, both sides of (13) are zero. By Assumption 1, the same level prior $\hat{\gamma}_r$ is used under π and $\overline{\pi}$. Joint convexity of $\operatorname{dist}(\cdot,\cdot)$, followed by (13), therefore gives

$$\begin{aligned}
\operatorname{dist}\left(\hat{T}^{(r)}_{h,\mathrm{agg}}(\mathcal{X};\pi), \hat{T}^{(r)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi})\right) &\le \sum_{v\in\mathcal{L}_r(\mathcal{T})} \hat{\gamma}_r(v)\, \operatorname{dist}\Big(\hat{T}_h(\mathcal{S}_v;\pi_v), \hat{T}_h(\mathcal{S}_v;\overline{\pi}_v)\Big) \\
&\le \varepsilon_{\mathrm{OC}} \sum_{v\in\mathcal{L}_r(\mathcal{T})} \hat{\gamma}_r(v)\, \operatorname{cay}(\pi_v, \overline{\pi}_v) \\
&= \kappa_r(\mathcal{T},\pi)\,\varepsilon_{\mathrm{OC}}.
\end{aligned} \tag{14}$$

Next, Lemma 9 bounds the discrepancy between the aggregates under canonical ordering:

$$\operatorname{dist}\left(\hat{T}^{(\ell)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi}), \hat{T}^{(\ell')}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi})\right) < (\ell' - \ell)\,\varepsilon_{\mathrm{SC}} + D_h \sum_{j=\ell}^{\ell'-1} \delta_j(\mathcal{T}). \tag{15}$$

Finally, insert these two canonical order aggregates between the original aggregates. By the triangle inequality, and the two bounds (14) and (15), we have

$$\begin{aligned}
\operatorname{dist}\left(\hat{T}^{(\ell)}_{h,\mathrm{agg}}(\mathcal{X};\pi), \hat{T}^{(\ell')}_{h,\mathrm{agg}}(\mathcal{X};\pi)\right) &\le \operatorname{dist}\left(\hat{T}^{(\ell)}_{h,\mathrm{agg}}(\mathcal{X};\pi), \hat{T}^{(\ell)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi})\right) \\
&\quad + \operatorname{dist}\left(\hat{T}^{(\ell)}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi}), \hat{T}^{(\ell')}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi})\right) \\
&\quad + \operatorname{dist}\left(\hat{T}^{(\ell')}_{h,\mathrm{agg}}(\mathcal{X};\overline{\pi}), \hat{T}^{(\ell')}_{h,\mathrm{agg}}(\mathcal{X};\pi)\right) \\
&< (\ell' - \ell)\,\varepsilon_{\mathrm{SC}} + D_h \sum_{j=\ell}^{\ell'-1} \delta_j(\mathcal{T}) + \left[\kappa_\ell(\mathcal{T},\pi) + \kappa_{\ell'}(\mathcal{T},\pi)\right] \varepsilon_{\mathrm{OC}},
\end{aligned}$$

which establishes the proof. □

Corollary 12 (Cross-tree consistency)**.** *Let $\mathcal{T}$ and $\mathcal{T}'$ be any two binary conditioning trees induced by $\mathcal{A} = \{A_1, \ldots, A_d\}$, rooted at the same population $\mathcal{X}$ with depths L and L', respectively. Assume $\mathcal{T}$ and $\mathcal{T}'$ are equipped with arbitrary constraint orderings π and π', respectively. Under the conditions of Theorem 11, for every $0 \le \ell \le L$ and $0 \le \ell' \le L'$ with $(\ell, \ell') \neq (0, 0)$, we have*

$$\begin{aligned}\operatorname{dist}\left(\hat{T}^{(\ell)}_{h,\mathrm{agg},\mathcal{T}}(\mathcal{X};\pi), \hat{T}^{(\ell')}_{h,\mathrm{agg},\mathcal{T}'}(\mathcal{X};\pi')\right) &< (\ell+\ell')\,\varepsilon_{\mathrm{SC}} + D_h\big[\Delta_\ell(\mathcal{T}) + \Delta_{\ell'}(\mathcal{T}')\big] \\ &\quad + \big[(\ell-1)_+ + (\ell'-1)_+\big]\,\varepsilon_{\mathrm{OC}},\end{aligned}$$

where $(x)_+ = \max\{x, 0\}$ and $\Delta_r(\mathcal{T}) \triangleq \sum_{j=0}^{r-1} \delta_j(\mathcal{T})$ defines the cumulative prior-refinement discrepancy through level r, with $\Delta_0(\mathcal{T}) \triangleq 0$.

Proof. The level-zero aggregate is the direct estimate at the common root and is therefore independent of the tree:

$$\hat{T}^{(0)}_{h,\mathrm{agg},\mathcal{T}}(\mathcal{X};\pi) = \hat{T}_h(\mathcal{X}) = \hat{T}^{(0)}_{h,\mathrm{agg},\mathcal{T}'}(\mathcal{X};\pi'). \tag{16}$$

For every $1 \le r \le L$, applying Theorem 11 within $\mathcal{T}$ to levels 0 and r and then invoking (11) gives

$$\begin{aligned}\operatorname{dist}\left(\hat{T}^{(r)}_{h,\mathrm{agg},\mathcal{T}}(\mathcal{X};\pi), \hat{T}_h(\mathcal{X})\right) &< r\,\varepsilon_{\mathrm{SC}} + D_h\,\Delta_r(\mathcal{T}) + \kappa_r(\mathcal{T},\pi)\,\varepsilon_{\mathrm{OC}} \\ &\le r\,\varepsilon_{\mathrm{SC}} + D_h\,\Delta_r(\mathcal{T}) + (r-1)_+\,\varepsilon_{\mathrm{OC}},\end{aligned} \tag{17}$$

The same bound holds within $\mathcal{T}'$ for every $1 \le r \le L'$. Now recall that $(\ell, \ell') \neq (0, 0)$. Using the direct root estimate as an intermediate point for the triangle inequality and applying (17) to each non-zero level, we obtain

$$\begin{aligned}&\operatorname{dist}\left(\hat{T}^{(\ell)}_{h,\mathrm{agg},\mathcal{T}}(\mathcal{X};\pi), \hat{T}^{(\ell')}_{h,\mathrm{agg},\mathcal{T}'}(\mathcal{X};\pi')\right) \\ &\qquad \le \operatorname{dist}\left(\hat{T}^{(\ell)}_{h,\mathrm{agg},\mathcal{T}}(\mathcal{X};\pi), \hat{T}_h(\mathcal{X})\right) + \operatorname{dist}\left(\hat{T}_h(\mathcal{X}), \hat{T}^{(\ell')}_{h,\mathrm{agg},\mathcal{T}'}(\mathcal{X};\pi')\right) \\ &\qquad < (\ell+\ell')\,\varepsilon_{\mathrm{SC}} + D_h\left[\Delta_\ell(\mathcal{T}) + \Delta_{\ell'}(\mathcal{T}')\right] + \left[(\ell-1)_+ + (\ell'-1)_+\right]\varepsilon_{\mathrm{OC}},\end{aligned}$$

where a term corresponding to a zero level vanishes by (16). This proves the claimed bound. □

D.3 Additional proof

Proof of Proposition 5. Consider any two level aggregates, arising at levels ℓ and ℓ', respectively, from BCTs induced by $\mathcal{A}$ and rooted at $\mathcal{X}$. If $\ell = \ell' = 0$, they coincide by (16). Otherwise, Corollary 12 gives

$$\begin{aligned}\operatorname{dist}\left(\hat{T}^{(\ell)}_{h,\mathrm{agg},\mathcal{T}}(\mathcal{X};\pi), \hat{T}^{(\ell')}_{h,\mathrm{agg},\mathcal{T}'}(\mathcal{X};\pi')\right) &< (\ell+\ell')\,\varepsilon_{\mathrm{SC}} + D_h\big[\Delta_\ell(\mathcal{T}) + \Delta_{\ell'}(\mathcal{T}')\big] \\ &\quad + \big[(\ell-1)_+ + (\ell'-1)_+\big]\varepsilon_{\mathrm{OC}} \\ &\le 2d\,\varepsilon_{\mathrm{SC}} + 2(d-1)D_h\,\varepsilon_{\mathrm{PR}} + 2(d-1)\,\varepsilon_{\mathrm{OC}},\end{aligned}$$

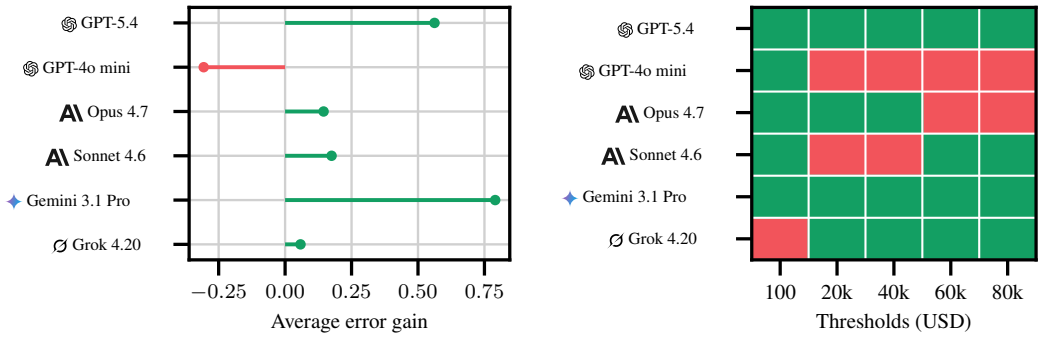
where the second inequality uses $\ell, \ell' \le d$, $\delta_0 = 0$, and $\Delta_r(\mathcal{T}) \le \max\{r-1, 0\}\,\varepsilon_{\mathrm{PR}}$. □

E Additional experimental results

This appendix reports additional analyses that probe the robustness of our experimental findings across prompting and implementation choices.

E.1 Partition-then-aggregate prompting

The aggregation experiments in Section 4 rely on an explicitly specified binary conditioning tree, where subgroup-level estimates from a fixed partition are recombined via the law of total probability. Their success raises a natural question: can the same benefit be recovered within a single query? We construct a prompt that instructs the model to first identify relevant subpopulations and then combine them into a population-level estimate. We call this strategy *partition-then-aggregate (PTA) prompting*. Unlike explicit tree-based aggregation, PTA prompting does not require an externally



(a) **Average error gain.** Average relative error gain of PTA prompting over direct prompting across the five income thresholds. Positive values indicate that PTA prompting reduces absolute error relative to direct prompting.

(b) **Win matrix.** Each tile corresponds to one model–threshold pair. Green indicates that PTA prompting yields lower absolute error than direct prompting; red indicates the complement.

Figure 10: **Benefit of PTA prompting on ACS income estimation.** We compare direct prompting with PTA prompting, which asks the model to first reason over relevant subpopulations before returning an aggregate prediction, while leaving the subpopulations implicit. Unlike explicit tree-based aggregation, the decomposition is not fixed in advance but chosen implicitly by the model. For each model and income threshold $\tau \in \{100, 20\text{k}, 40\text{k}, 60\text{k}, 80\text{k}\}$ USD, we evaluate the absolute error between the predicted population-level probability $\mathbb{P}(Y > \tau \,|\, X \in \mathcal{X})$ and the ACS survey-based ground truth.

specified partition. If successful, this would provide a lightweight and broadly applicable prompting strategy. The exact prompt template is provided in Appendix H.4.

To evaluate this idea, we compare PTA prompting against a direct prompting baseline, in which the model is asked to estimate the aggregate target quantity in a single step without any reference to subpopulations. On the ACS income task, Figure 10a shows that PTA prompting often improves over direct prompting when averaged across the five income thresholds. However, the gains are model-dependent and less systematic than those obtained with explicit tree-based aggregation. PTA prompting therefore recovers part of the benefit of decomposition through a lightweight implicit strategy, but it is not uniformly beneficial across models. A more detailed analysis of when model-chosen decompositions help is left for future work.

E.2 Self-consistency across problem domains

This subsection reports the additional self-consistency results. We first vary the prediction target within the ACS setting and then move to opinion distributions from the World Values Survey.

ACS prediction tasks. First, we evaluate self-consistency on the ACS setting discussed in the main body. The ACS income prediction task corresponds to a widely used testbed for evaluating alignment and fairness of predictive models [Ding et al., 2021, Cruz et al., 2024]. We take $\mathcal{A}$ to be the demographic attributes underlying the tree in Section 3 and, for every conditioning event induced by $\mathcal{A}$, elicit the model's estimated binned income distribution. We then apply the self-consistency checks from Section 5. The two columns in Table 2a report the two complementary benchmark scores. The first is the split consistency score $\mathrm{SC}_\varepsilon(\mathcal{A})$. For every admissible pair of a conditioning event and a next split attribute in $\mathcal{C}_{\mathrm{SC}}(\mathcal{A})$, we compare the direct estimate to the prior-weighted aggregate of its two refinements and report the fraction of checks satisfied at tolerance $\varepsilon = 0.02$. The second is the order consistency score $\mathrm{OC}_\varepsilon(\mathcal{A})$. For every two-attribute conditioning event in $\mathcal{C}_{\mathrm{OC}}(\mathcal{A})$, we elicit direct estimates under both orderings of the attribute constraints and report the fraction of pairs whose predictions agree within tolerance $\varepsilon = 0.02$. Across frontier models, we observe widespread violations of both criteria. These discrepancies point to an important weakness of model-induced estimates and help explain the varying alignment scores of reconstructed estimates across levels.

We next hold the population structure fixed while changing the prediction target. In the binned ACS commute time task reported in Table 2b, we use the same conditioning attributes and subgroup priors

Model	Split consistency	Order consistency
GPT-5.4	0.33	1.00
Sonnet 4.6	0.00	0.25
Grok 4.3	0.00	0.50
Qwen 3.6 Plus	0.17	0.50

(a) **ACS income prediction.** The target variable Y is the binned income distribution.

Model	Split consistency	Order consistency
GPT-5.4	0.17	0.50
Sonnet 4.6	0.17	0.50
Grok 4.3	0.17	0.50
Qwen 3.6 Plus	0.33	0.25

(b) **ACS commute time prediction.** The target variable Y is the binned commute time distribution.

Table 2: **Self-consistency in ACS prediction tasks.** We evaluate self-consistency for a fixed tree and two different prediction tasks. The attribute set $\mathcal{A}$ consists of age and employment status as used in Figure 14. For each task, we report the split consistency score $\mathrm{SC}_\varepsilon(\mathcal{A})$ and the order consistency score $\mathrm{OC}_\varepsilon(\mathcal{A})$ at tolerance $\varepsilon = 0.02$. Higher values indicate stronger self-consistency and are shown in lighter red tones. We observe widespread violations of self-consistency across frontier models.

	Split consistency											**Order consistency**										
	Canada					Indonesia						Canada					Indonesia					
Model	Q_a	Q_b	Q_c	Q_d	Q_e	Q_a	Q_b	Q_c	Q_d	Q_e	Avg.	Q_a	Q_b	Q_c	Q_d	Q_e	Q_a	Q_b	Q_c	Q_d	Q_e	Avg.
GPT-5.6 Sol	1.00	1.00	0.83	1.00	0.83	0.67	0.83	0.67	1.00	0.50	0.83	0.75	0.75	0.25	0.75	1.00	0.50	1.00	1.00	1.00	1.00	0.80
GPT-5.4	1.00	1.00	0.33	0.50	0.33	0.67	0.50	0.50	0.33	0.33	0.55	1.00	0.75	0.25	0.75	0.50	1.00	1.00	0.75	0.25	0.75	0.70
GPT-5.4 nano	0.33	0.17	0.17	0.00	0.33	0.17	0.17	0.17	0.17	0.33	0.20	0.25	0.75	0.00	0.00	0.50	0.50	0.25	0.50	0.25	0.25	0.32
GPT-5.1	0.67	0.67	0.33	0.33	0.50	0.83	0.50	0.33	0.33	0.50	0.50	0.50	1.00	0.75	1.00	0.75	0.25	0.00	0.75	1.00	0.50	0.65
GPT-4o mini	0.83	0.00	0.50	0.00	0.33	0.83	0.33	0.50	0.50	0.33	0.42	0.25	0.50	0.50	0.25	1.00	0.50	1.00	0.25	0.25	0.25	0.47
Sonnet 4.6	0.67	1.00	0.50	1.00	0.67	0.50	0.83	1.00	0.50	0.33	0.70	1.00	0.75	1.00	1.00	1.00	0.50	1.00	0.75	1.00	1.00	0.90
DeepSeek-V3.2	0.50	0.17	0.17	0.17	0.17	0.17	0.83	0.50	0.17	0.33	0.32	0.75	0.50	0.25	0.00	0.25	0.50	0.50	0.25	0.00	0.50	0.35
Qwen 3.6 Plus	0.33	1.00	0.33	0.50	0.17	0.33	0.67	0.67	0.83	0.17	0.50	0.75	0.75	0.75	0.50	0.50	0.75	0.50	0.25	0.50	0.50	0.57
Qwen 3.5 27B	0.67	0.17	0.50	0.50	0.17	0.00	0.33	0.50	0.33	0.00	0.32	0.25	0.75	0.25	0.50	0.00	0.50	0.25	0.25	0.50	0.75	0.40
Qwen 3.5 35B	0.50	0.50	0.33	0.50	0.00	0.17	0.67	0.33	0.67	0.17	0.38	0.25	0.75	0.25	0.50	0.50	0.50	0.25	0.25	0.25	0.50	0.40
Qwen 3.5 122B	0.50	0.33	0.50	0.33	0.17	0.33	0.67	0.50	0.33	0.33	0.40	0.00	1.00	0.50	0.75	0.50	0.50	0.50	0.50	0.75	0.50	0.55
Grok 4.5	0.67	1.00	0.50	0.67	0.67	0.17	1.00	0.83	0.50	0.50	0.65	1.00	1.00	1.00	0.75	0.50	0.75	0.75	1.00	1.00	1.00	0.88
Grok 4.3	0.67	0.83	0.17	0.33	0.17	0.83	0.83	0.17	0.33	0.50	0.48	0.25	0.00	1.00	0.50	0.50	0.50	1.00	1.00	0.25	0.50	0.55
Grok 4.1 Fast	0.50	0.50	0.00	0.83	0.17	0.67	1.00	0.67	0.17	0.33	0.48	0.50	0.75	0.75	1.00	0.50	0.25	0.75	1.00	1.00	0.25	0.68
Gemini 3 Flash	0.33	1.00	0.33	0.83	0.50	0.17	0.83	0.83	0.33	1.00	0.62	1.00	0.75	0.50	1.00	0.75	1.00	0.75	0.25	0.75	1.00	0.78

Table 3: **Self-consistency in global opinion prediction.** For five survey questions sampled from the WVS $(Q_a, \ldots, Q_e)$ and two countries, we report the split consistency score $\mathrm{SC}_\varepsilon(\mathcal{A})$ and the order consistency score $\mathrm{OC}_\varepsilon(\mathcal{A})$ at tolerance $\varepsilon = 0.02$, where $\mathcal{A}$ consists of age (younger than 45 years old vs. at least 45 years old) and income splits (low vs. high). Each survey question defines a distinct prediction problem. Higher values indicate stronger self-consistency and are shown in lighter red tones. Across models, countries, and questions, current models fail to satisfy the self-consistency checks reliably.

as in the income experiment, but ask models to estimate a different target distribution. The resulting split and order consistency scores differ substantially from those obtained for income, showing that self-consistency is not solely a property of the model or conditioning tree, but can also vary across prediction tasks.

Global opinion modeling. The split and order consistency scores cover all conditioning trees induced by a fixed attribute set $\mathcal{A}$ (see Proposition 5). A consistent conditional estimator, however, must satisfy these identities not only across trees, but for every attribute set and every problem domain. To probe this problem axis, we construct binary conditioning trees from the World Values Survey (WVS), extending our evaluation to opinion distributions across countries beyond the United States. Building on the GlobalOpinionQA dataset [Durmus et al., 2024], we use demographic attributes as conditioning variables and examine, for each country, how subpopulation-level estimates aggregate to the corresponding country-level estimate. Each survey question thereby defines a distinct prediction problem on which the consistency checks are instantiated separately.

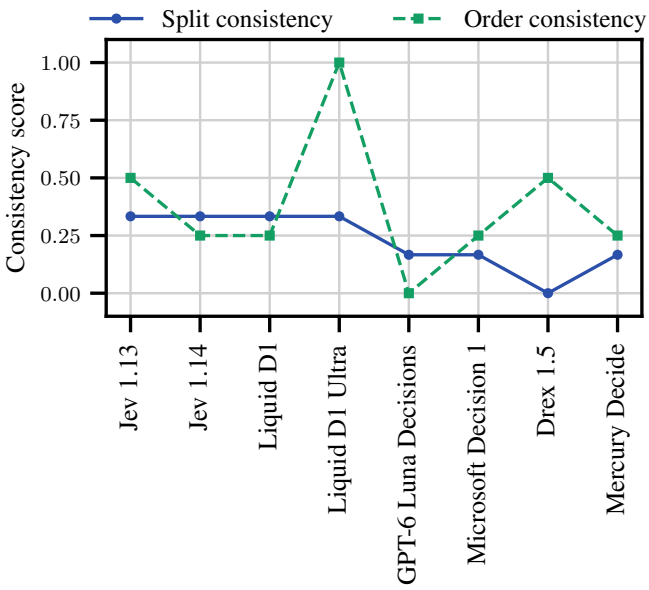


Figure 11: **Decision models violate probabilistic self-consistency.** We report the split consistency score $\mathrm{SC}_\varepsilon(\mathcal{A})$ and order consistency score $\mathrm{OC}_\varepsilon(\mathcal{A})$ at tolerance $\varepsilon = 0.02$ for the thresholded ACS income prediction task with $\tau = 40\text{k}$ USD, following the experimental setting of Figure 4. Despite training specifically for decision tasks and calibrated probability estimation, these models fail basic consistency checks across their own predictions.

We run our evaluation on data from Canada and Indonesia, the two countries with the largest sample sizes. For each country, we construct a BCT of depth two using age and income splits. We then elicit categorical response distributions at every node for five selected survey questions. The full question wording and answer scales are reported in Appendix J. Consistency is assessed using the reference-free checks from Section 5. For each question, this includes the split and order consistency checks. Results for tolerance $\varepsilon = 0.02$ are reported in Table 3.

The WVS results show that the self-consistency failures observed on ACS are not specific to income prediction or to the U.S. census setting. Across both countries, models satisfy only a fraction of the split and order consistency checks. No model is uniformly consistent across questions. Even the strongest models exhibit scores that vary substantially across survey questions and countries. Thus, the WVS provides complementary evidence that current models do not elicit faithful probability estimates.

E.3 Self-consistency in decision models

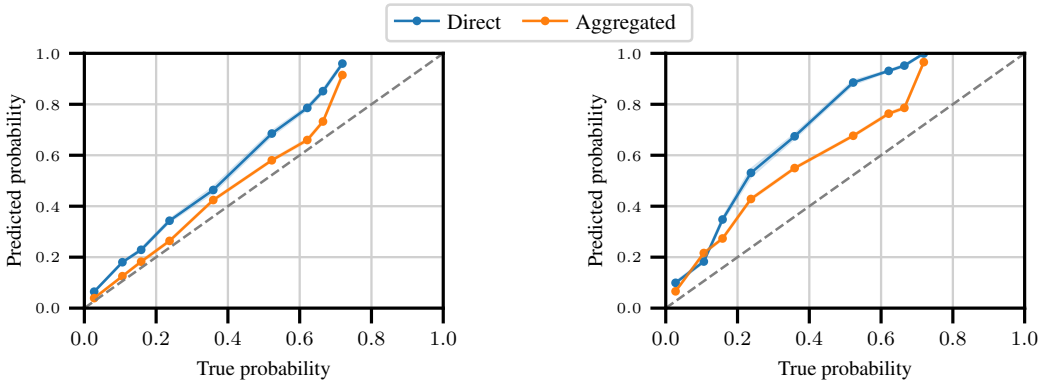
We extend the evaluation in Figure 4 to decision models, which are trained specifically for decision tasks and calibrated probability estimation. We retain the thresholded ACS income prediction task with $\tau = 40\text{k}$ USD, demographic attribute set $\mathcal{A}$, and consistency checks. Decision models require a predefined structured output format, which we provide in Appendices H.8 and H.9 together with the prompt and context. We evaluate split and order consistency at tolerance $\varepsilon = 0.02$. Figure 11 shows that, despite their explicit focus on reliable probability estimation, these models violate basic probabilistic consistency requirements. Structured outputs and calibration-oriented training thus leave a fundamental reliability gap.

E.4 Level-wise self-consistency

To assess how self-consistency varies with the granularity of the partition, Table 4 reports a level-wise root consistency evaluation. For this experiment, we use the first four levels of the depth-six binary conditioning tree described in Appendix F.3. For each level ℓ, we reconstruct the aggregate income distribution from the partition induced at that level and compare it to the model's direct root estimate

Model	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$
GPT-5.4	0.084	0.145	0.070	0.050
Sonnet 4.6	0.042	0.108	0.064	0.053
Grok 4.3	0.060	0.116	0.081	0.063
Qwen 3.6 Plus	0.045	0.143	0.079	0.121

Table 4: **Level-wise root consistency.** For each level ℓ of a depth-four tree, we reconstruct the aggregate income distribution from the corresponding level-ℓ partition and compare it to the model's direct root estimate. Entries report the normalized Wasserstein-1 distance. Lower values indicate stronger agreement with the root estimate and are highlighted using darker green tones.



(a) **Sociodemographic prompting.** The target subpopulation is described in the third person using structured attributes.

(b) **Persona prompting.** The model is instructed to role-play as a member of the target subpopulation in the first person.

Figure 12: **Prompt sensitivity of probability elicitation.** Results are shown for GPT-5.4 without reasoning, using ground-truth priors and 100 repeated LLM samples. For multiple income thresholds, model estimates are plotted against the corresponding survey-derived probabilities. Points closer to the diagonal indicate stronger ground-truth alignment. Direct estimates are obtained from population-level prompts, whereas reconstructed estimates are obtained by aggregating subgroup-level estimates via the law of total probability using ground-truth priors. The population is decomposed according to employment status; see the ESR attribute in Appendix I. Error bars correspond to 90 % confidence intervals computed by bootstrapping with 1000 samples. Error bars are barely visible as the intervals are very narrow.

using the normalized Wasserstein distance $\overline{\mathcal{W}}_1$. The table reports these discrepancies directly, so lower values indicate stronger agreement with the root estimate.

E.5 Prompt sensitivity

We examine prompt sensitivity along two complementary dimensions: ground-truth alignment and reference-free self-consistency. In the default *sociodemographic* formulation, the target subpopulation is described in the third person through a structured list of attributes; see the prompt template in Appendix H.2. In the *persona prompting* formulation, the same information is instead presented as a first-person role-playing instruction; see Appendix H.3. Both formulations elicit the same verbalized probability estimate for the same event and differ only in how the conditioning information is expressed.

We first assess ground-truth alignment in the thresholded ACS income task. Across the two prompt formulations, we keep the model, sampling temperature, number of repeated generations, outcome event, income thresholds, and aggregation weights fixed. The resulting comparisons with the survey-

	Split consistency											Order consistency										
	Canada					Indonesia						Canada					Indonesia					
Model	Q_a	Q_b	Q_c	Q_d	Q_e	Q_a	Q_b	Q_c	Q_d	Q_e	Avg.	Q_a	Q_b	Q_c	Q_d	Q_e	Q_a	Q_b	Q_c	Q_d	Q_e	Avg.
GPT-5.4	0.17	0.33	0.17	0.50	0.33	0.17	0.17	0.67	0.00	0.17	0.27	0.75	1.00	0.75	1.00	0.75	0.00	0.75	0.75	1.00	0.75	0.75
Sonnet 4.6	0.83	0.67	0.33	0.67	0.33	0.17	0.50	0.50	0.17	0.50	0.47	1.00	1.00	1.00	1.00	0.50	0.75	0.75	0.75	1.00	0.50	0.82
Qwen 3.6 Plus	0.50	0.83	0.17	1.00	0.33	0.00	0.83	0.33	0.67	0.33	0.50	0.75	0.50	0.75	0.00	0.25	0.75	0.75	0.25	0.50	0.50	0.50
Qwen 3.5 27B	0.67	1.00	0.50	1.00	0.17	0.00	0.33	0.50	0.33	0.33	0.48	0.25	0.50	0.50	0.00	0.25	0.50	0.50	0.50	0.50	0.75	0.43
Qwen 3.5 35B	0.50	0.50	0.33	0.50	0.00	0.33	0.67	0.67	0.33	0.17	0.40	0.25	0.50	0.25	0.75	0.25	0.50	0.75	0.50	0.75	0.50	0.50
Qwen 3.5 122B	0.33	0.33	0.33	0.50	0.00	0.17	0.00	0.33	0.83	0.17	0.30	0.25	0.50	1.00	0.50	0.75	0.50	0.50	0.25	0.75	0.50	0.55

Table 5: **Robustness to prompt formulation.** Self-consistency results under first-person persona prompting, using the same experimental setting, notation, and evaluation criteria as Table 3. Higher values indicate stronger self-consistency. The effects vary across models, and substantial violations persist.

derived probabilities are shown in Figure 12. Under both formulations, estimates reconstructed through the law of total probability lie closer to the diagonal than direct population-level estimates for most thresholds. The benefit of subgroup decomposition is therefore not specific to a particular formulation of the conditioning context. However, persona prompting systematically overestimates the ground-truth probabilities, whereas sociodemographic prompting produces more closely aligned estimates. We therefore use sociodemographic prompting as the default throughout our experiments.

We next examine the complementary, reference-free question of whether the elicited estimates are statistically self-consistent. For a subset of models, we repeat the WVS evaluation from Table 3, changing only the prompting method. Specifically, we replace the default third-person sociodemographic template with the first-person persona template. Table 5 reports the fraction of self-consistency checks satisfied by each model, revealing that the effect of persona prompting varies across models. It lowers self-consistency for some models while improving it for others. Averaged over the six models evaluated under both prompt formulations, the split consistency rate decreases from 0.475 under sociodemographic prompting to 0.403 under persona prompting. The order consistency rate increases slightly from 0.587 to 0.592. Thus, although prompt formulation affects individual consistency rates, substantial violations persist under both interfaces, leaving our central qualitative finding unchanged.

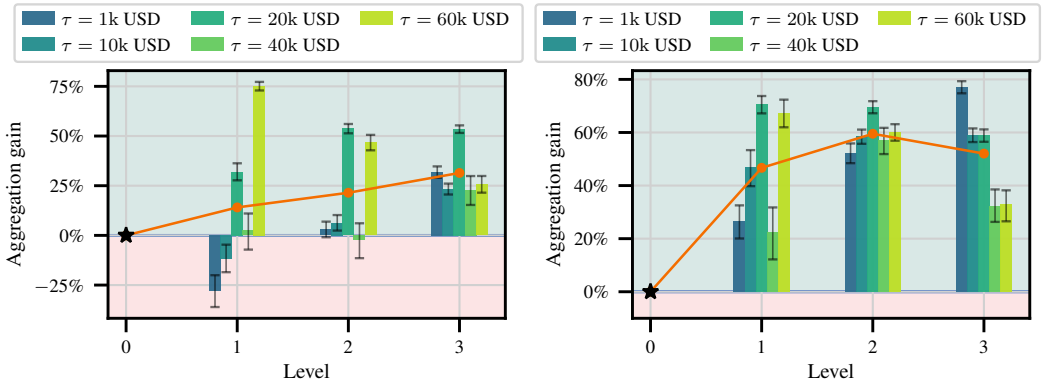
E.6 Sensitivity to split verbalization

We examine whether the aggregation gains reported in Section 4 depend on how the conditioning-attribute splits are expressed in natural language. To isolate this effect, we repeat the thresholded ACS income experiment while varying only the textual descriptions of the first three splits. The underlying partitions remain unchanged. We compare the default verbalization, which explicitly enumerates the attribute values assigned to each branch, with a simplified verbalization that expresses the same sets using compact intervals or semantically meaningful category descriptions. Specifically, at level one, the `AGEP` branches are described using age ranges. For example, the explicit enumeration "31 years old or 32 years old or ..." is replaced by "31 to 68 years old." At level two, the `COW` values $\{1, \ldots, 7\}$ are summarized as "currently employed or self-employed," whereas values $\{8, 9, \texttt{n/a}\}$ are summarized as "not currently in paid employment." These groupings preserve the ACS category mapping reported in Table 7. Finally, at level three, the `WKHP` split is expressed using compact intervals, such as "1 to 39 hours per week," instead of explicitly listing every constituent value.

As shown in Figure 13, changing the split descriptions affects the magnitude of the resulting aggregation gains. Nevertheless, the central qualitative pattern remains unchanged: estimates reconstructed from finer-grained subgroup predictions generally improve upon direct population-level prompting under both verbalization schemes. These results show that the macro fallacy is not an artifact of the split verbalization method.

E.7 Robustness to reasoning effort

We assess robustness to reasoning effort by repeating the self-consistency evaluation from Table 3 under the same experimental setup, but instructing the models to use a "low" reasoning budget. We compare these results with the corresponding runs without reasoning while keeping all other



(a) **Default verbalization.** The attribute values defining each branch are enumerated explicitly.

(b) **Simplified verbalization.** The same partitions are described using compact intervals and natural-language category summaries.

Figure 13: **Sensitivity of aggregation gains to split verbalization.** We report the aggregation gain across the first three levels of the thresholded ACS income tree under the default and simplified verbalization schemes. All experimental choices other than the subgroup descriptions are held fixed. Positive values indicate that estimates reconstructed from subgroup predictions are better aligned with the survey-derived reference than the direct population-level estimate at level 0. Bars correspond to different income thresholds, and the orange curve shows their average. Error bars denote 90 % bootstrap confidence intervals. Aggregation gains persist under both schemes and are more uniformly positive under the simplified descriptions.

	Split consistency											Order consistency										
	Canada					Indonesia						Canada					Indonesia					
Model	Q_a	Q_b	Q_c	Q_d	Q_e	Q_a	Q_b	Q_c	Q_d	Q_e	Avg.	Q_a	Q_b	Q_c	Q_d	Q_e	Q_a	Q_b	Q_c	Q_d	Q_e	Avg.
GPT-5.4	0.50	1.00	0.50	1.00	0.33	0.50	0.83	0.33	0.50	0.33	0.58	1.00	0.25	1.00	0.75	0.75	0.75	0.50	0.75	0.50	1.00	0.72
GPT-5.4 nano	0.33	0.50	0.00	0.67	0.33	0.00	0.17	0.83	0.17	0.33	0.33	0.75	0.50	0.50	0.75	0.00	0.00	0.50	0.50	0.50	0.50	0.45
GPT-4o mini	0.33	0.17	0.50	0.33	0.50	0.83	0.50	0.17	0.83	0.33	0.45	1.00	0.75	1.00	0.50	0.25	0.50	0.75	0.75	0.50	0.50	0.65
Sonnet 4.6	0.33	1.00	0.33	0.33	0.50	0.33	1.00	0.83	0.33	0.33	0.53	0.50	0.75	0.75	1.00	0.50	1.00	1.00	1.00	0.25	0.50	0.72
Gemini 3 Flash	0.33	1.00	0.67	0.83	0.33	0.17	0.83	1.00	0.83	0.67	0.67	1.00	1.00	0.50	0.25	0.75	0.75	0.50	0.75	0.75	1.00	0.72

Table 6: **Robustness to reasoning effort.** Self-consistency results obtained with a low reasoning budget, using the same experimental setting, notation, and evaluation criteria as Table 3. Higher values indicate stronger self-consistency. Additional reasoning does not reliably eliminate self-consistency violations.

evaluation choices fixed. Because explicit reasoning controls are not available for every model, the analysis is restricted to models for which both settings can be evaluated. Table 6 reports the fraction of self-consistency checks satisfied by each model.

Enabling reasoning affects individual model performance, but the effects vary considerably across models. Averaged across the evaluated models, the split consistency rate increases only modestly, from 0.498 without reasoning to 0.512 with a low reasoning budget. The order consistency rate exhibits a similarly small increase, from 0.634 to 0.652. Thus, additional reasoning does not reliably eliminate the observed statistical consistency violations. Our central qualitative finding is therefore robust to enabling low-effort reasoning.

F Experiment details

In this appendix, we describe the experimental setup[6] used for the results reported throughout this paper.

[6]The code is available at: https://github.com/patrikwolf/statistical_self_consistency

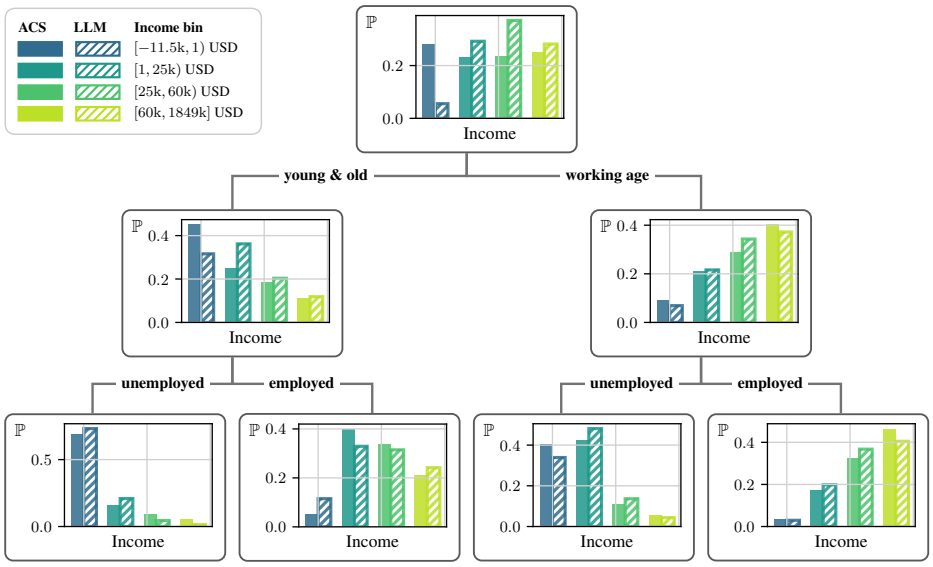
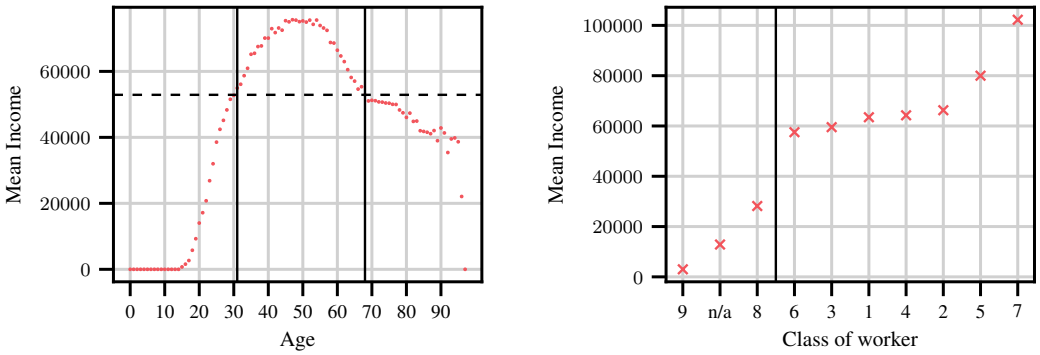


Figure 14: **Ground-truth and LLM-elicited income distributions across partitions of the population.** We construct a binary conditioning tree from the 2024 ACS population by recursively slicing the population with sociodemographic attributes. The first split criterion corresponds to age (AGEP) and the second to class of worker (COW). Each node reports the distribution of yearly income over four bins. For each bin, the solid bar shows the survey-based ground-truth distribution, while the hatched bar shows the LLM-estimated conditional distribution.



(a) Age attribute AGEP.

(b) Class-of-worker attribute COW.

Figure 15: **Mean income across attribute-induced subgroups.** We partition the ACS population by all attainable values of a single attribute and report the survey-weighted mean yearly income in USD for each resulting subgroup. For the age attribute in Figure 15a, subgroup values are displayed in their natural order, and the binary split is defined by thresholding the corresponding subgroup mean incomes (dashed line). For the categorical class-of-worker attribute in Figure 15b, subgroup values are first sorted by mean income before defining the split. Both rules yield splits that separate lower-income from higher-income attribute values while keeping the two branches approximately balanced in survey-weighted probability mass.

F.1 Binary conditioning tree

For our running example in Section 3, we construct the binary conditioning tree shown in Figure 14 using demographic attributes from the 2024 American Community Survey (ACS) 1-Year Public Use Microdata Sample [U.S. Census Bureau, 2024]. The root node corresponds to the U.S. population, which we recursively refine using two binary splits. To select these splits, we consider the following candidate attributes: AGEP (age), CIT (citizenship), COW (class of worker), ESR (employment status), MAR (marital status), OCCP (occupation), SCHL (educational attainment), SEX (sex), and WKHP (usual

hours worked per week). For each candidate attribute, we first partition the population by all attainable attribute values and compute the mean yearly income within each resulting subgroup, as exemplified in Figure 15. We then group the attribute values into a lower-income and a higher-income branch, choosing the grouping so that the two branches have approximately equal survey-weighted probability mass. Finally, we qualitatively select the two attributes whose induced binary splits yield particularly distinctive income distributions across the two branches. This procedure yields the two splits used in our main ACS experiments. The first split is induced by the ACS age attribute `AGEP`. Respondents younger than 31 or older than 68 are grouped into the "young & old" subgroup, while respondents aged 31 to 68 are grouped into the "working age" subgroup. The second split uses the class-of-worker attribute `COW`. Values 1–7 are labeled as "employed," whereas values 8, 9, and missing values are mapped to "unemployed." The corresponding ACS attribute descriptions are reported in Table 7.

At each node in Figure 14, we report the survey-weighted distribution of yearly income in USD. For the four-bin income setting, the bin edges are chosen so that the root distribution is approximately uniform, and the same binning is then used for all subpopulations in the tree. Since yearly income is unavailable for respondents below age 15, we set their income to 0 USD in our experiments, reflecting the assumption that individuals in this age group have no reported labor income.

F.2 Prior elicitation

To obtain fully model-induced aggregate quantities, both the subgroup-conditional estimates and the subgroup priors must be elicited from the LLM. In this subsection, we describe the level-wise procedure used to estimate these priors. For each tree level ℓ with induced partition $\mathcal{P}_\ell$, we prompt the model $n = 50$ times to estimate the probability mass that a randomly sampled individual from the base population belongs to each subgroup $\mathcal{S} \in \mathcal{P}_\ell$. The prompts use the template in Appendix H.1, which lists the subgroups in the partition and asks the model to return a normalized probability distribution over them. Although the prompt instructs the model to output probabilities that sum to one, the raw responses may contain small normalization errors. We therefore post-process each response before averaging. The LLM-estimated prior at level ℓ is then obtained by averaging the renormalized estimates across repetitions.

F.3 Alignment error

The alignment analysis in Figure 2 is based on a binary conditioning tree over ACS attributes. The tree is defined by the following sequence of splits:

Level 1: `AGEP` is split into `{0, ..., 30, 69, ..., 96, n/a}` and `{31, ..., 68}`.

Level 2: `COW` is split into `{1, ..., 7}` and `{8, 9, n/a}`.

Level 3: `WKHP` is split into `{1, ..., 39, n/a}` and `{40, ..., 99}`.

Level 4: `OCCP` is split into `{2060, 2300, 2740, 3401, 3421, 3422, 3424, 3603, 3605, 3620, 3630, 3640, 3645, 3647, 3648, 3649, 3940, 3960, 2350, 2440, 2545, 2633, 2723, 3430, 3601, 3602, 3646, 3946, 4010, 4020, 4030, 4040, 4055, 4110, 4120, 4130, 4140, 4150, 4160, 4220, 4230, 4251, 4255, 4350, 4420, 4435, 4461, 4500, 4510, 4521, 4522, 4525, 4530, 4540, 4600, 4621, 4622, 4640, 4655, 4720, 4740, 4760, 4900, 4940, 5020, 5160, 5260, 5300, 5310, 5320, 5400, 5510, 5610, 5810, 5820, 5850, 5860, 5900, 6040, 6050, 6120, 6410, 6600, 7260, 7510, 7610, 7800, 7810, 7830, 7840, 7850, 8256, 8300, 8310, 8320, 8335, 8350, 8365, 8465, 8530, 8540, 8710, 8800, 8850, 8910, 8950, 9110, 9142, 9350, 9365, 9415, 9600, 9610, 9620, 9630, 9640, 9645, 9720, 9920, n/a}` and `{10, 20, 40, 51, 52, 60, 101, 102, 110, 120, 135, 136, 137, 140, 150, 160, 205, 220, 230, 300, 310, 335, 340, 350, 360, 410, 420, 425, 440, 500, 510, 520, 530, 540, 565, 600, 630, 640, 650, 700, 705, 710, 725, 726, 735, 750, 800, 810, 820, 830, 845, 850, 860, 900, 910, 930, 940, 960, 1005, 1006, 1007, 1010, 1021, 1022, 1031, 1032, 1050, 1065, 1105, 1106, 1108, 1200, 1220, 1240, 1305, 1306, 1310, 1320, 1340, 1350, 1360, 1400, 1410, 1420, 1430, 1440, 1450, 1460, 1520, 1530, 1541, 1545, 1551, 1555, 1560, 1600, 1610, 1640, 1650, 1700, 1710, 1720, 1745, 1750, 1760, 1800, 1821, 1822,`

1825, 1840, 1860, 1900, 1910, 1920, 1935, 1970, 1980, 2001, 2002, 2003, 2004, 2005, 2006, 2011, 2012, 2013, 2014, 2015, 2016, 2025, 2040, 2050, 2100, 2105, 2145, 2170, 2180, 2205, 2310, 2320, 2330, 2360, 2400, 2435, 2555, 2600, 2631, 2632, 2634, 2635, 2636, 2640, 2700, 2710, 2721, 2722, 2751, 2752, 2755, 2770, 2805, 2810, 2825, 2830, 2840, 2850, 2861, 2862, 2865, 2905, 2910, 2920, 3000, 3010, 3030, 3040, 3050, 3090, 3100, 3110, 3120, 3140, 3150, 3160, 3200, 3210, 3220, 3230, 3245, 3250, 3255, 3256, 3258, 3261, 3270, 3300, 3310, 3321, 3322, 3323, 3324, 3330, 3402, 3423, 3500, 3515, 3520, 3545, 3550, 3610, 3655, 3700, 3710, 3720, 3725, 3740, 3750, 3801, 3802, 3820, 3840, 3870, 3900, 3910, 3930, 3945, 4000, 4200, 4210, 4240, 4252, 4330, 4340, 4400, 4465, 4700, 4710, 4750, 4800, 4810, 4820, 4830, 4840, 4850, 4920, 4930, 4950, 4965, 5000, 5010, 5040, 5100, 5110, 5120, 5140, 5150, 5165, 5220, 5230, 5240, 5250, 5330, 5340, 5350, 5360, 5410, 5420, 5500, 5521, 5522, 5530, 5540, 5550, 5560, 5600, 5630, 5710, 5720, 5730, 5740, 5840, 5910, 5920, 5940, 6005, 6010, 6115, 6130, 6200, 6210, 6220, 6230, 6240, 6250, 6260, 6305, 6330, 6355, 6360, 6400, 6441, 6442, 6460, 6515, 6520, 6530, 6540, 6660, 6700, 6710, 6720, 6730, 6740, 6765, 6800, 6825, 6835, 6850, 6950, 7000, 7010, 7020, 7030, 7040, 7100, 7120, 7130, 7140, 7150, 7160, 7200, 7210, 7220, 7240, 7300, 7315, 7320, 7330, 7340, 7350, 7360, 7410, 7420, 7430, 7540, 7560, 7640, 7700, 7720, 7730, 7740, 7750, 7855, 7905, 7925, 7950, 8000, 8025, 8030, 8040, 8100, 8130, 8140, 8225, 8250, 8255, 8450, 8500, 8510, 8555, 8600, 8610, 8620, 8630, 8640, 8650, 8720, 8730, 8740, 8750, 8760, 8810, 8830, 8920, 8930, 8940, 8990, 9005, 9030, 9040, 9050, 9121, 9122, 9130, 9141, 9150, 9210, 9240, 9265, 9300, 9310, 9410, 9430, 9510, 9570, 9650, 9760, 9800, 9810, 9825, 9830}.

Level 5: ESR is split into {1, 2, 4, 5} and {3, 6, n/a}.

Level 6: MAR is split into {1, 3} and {2, 4, 5, n/a}.

The ACS category mapping is reported in Table 7. For each level $\ell \in \{0, 1, \ldots, 6\}$, we compute the alignment error $\mathrm{AErr}(\ell)$ of the aggregate estimate obtained from the nodes at that level. To compare errors across levels, we normalize by the root-level error and report $1 - \mathrm{AErr}(\ell)/\mathrm{AErr}(0)$. Here, $\ell = 0$ corresponds to direct prompting, while larger values of ℓ correspond to estimates reconstructed from increasingly fine-grained subgroup predictions. Error bars indicate 90 % confidence intervals computed using 1000 bootstrap samples. In Figure 2a, we compare a selection of models at a fixed income threshold of $\tau = 40\mathrm{k}$ USD. In Figure 2b, we show the aggregation gain for GPT-5.4 across income thresholds. Across both settings, the results show that aggregate estimates reconstructed from subgroup estimates improve over direct root-level estimates up to moderate tree depths.

F.4 Disentangling the sources of aggregation error

The two panels in Figures 3a and 3b are based on the depth-six binary conditioning tree described in Appendix F.3. For visual clarity, we display only the first three levels in the conditional-estimates panel and the first four levels in the subgroup-priors panel.

Node-wise conditional alignment. For Figure 3a, we evaluate the quality of the LLM-estimated conditional income distributions at each node of the tree. The outcome is the four-bin income distribution used in Figure 14. For every node $\mathcal{S}$, we compare the LLM-estimated conditional distribution $\hat{T}_h(\mathcal{S})$ to the corresponding survey-based conditional distribution $T_h(\mathcal{S})$ using the normalized Wasserstein distance, $\overline{\mathcal{W}}_1\big(\hat{T}_h(\mathcal{S}), T_h(\mathcal{S})\big)$. The node color encodes this error. The resulting plot illustrates that conditioning on more specific subgroups improves conditional alignment relative to the root estimate, but that the improvement is heterogeneous across the tree. Some fine-grained subgroups with negligible prior mass remain poorly aligned.

Prior alignment. For Figure 3b, we evaluate the quality of the subgroup priors. For each tree level ℓ, let γ_ℓ denote the survey-based distribution over the partition $\mathcal{P}_\ell(\mathcal{X})$, and let $\hat{\gamma}_\ell$ denote the corresponding LLM-estimated prior distribution obtained with the elicitation procedure from

Appendix F.2. We measure prior alignment using total variation distance, $\mathrm{TV}(\hat{\gamma}_\ell, \gamma_\ell)$. This yields one prior-alignment error per model and tree level. The results show that prior estimation becomes increasingly difficult as the tree is refined. Coarse partitions are estimated comparatively accurately, whereas deeper levels require the model to distribute probability mass across a larger number of more specific subpopulations. Thus, aggregation involves a trade-off. More specific conditioning can improve the subgroup outcome estimates, but the corresponding population weights become harder to estimate accurately.

F.5 Variance decomposition across tree levels

The variance decomposition in Figure 3c is computed on the first four levels of the binary conditioning tree described in Appendix F.3. For each node, we compute the survey-weighted subgroup prior $\gamma_{\mathcal{S}}$, the conditional mean $\mu_{\mathcal{S}} \triangleq \mathbb{E}[Y \,|\, X \in \mathcal{S}]$, and the conditional variance $\mathbb{V}(Y \,|\, X \in \mathcal{S})$ from the ACS data. We then decompose the aggregate income variance into a within-subgroup and a cross-subgroup component. The within-subgroup term is the prior-weighted average conditional variance,

$$V_{\mathrm{within}}(\ell) \triangleq \sum_{\mathcal{S} \in \mathcal{P}_\ell(\mathcal{X})} \gamma_{\mathcal{S}}\, \mathbb{V}(Y \,|\, X \in \mathcal{S}),$$

which measures the residual heterogeneity that remains after conditioning on the attributes defining the level-ℓ subgroups. The cross-subgroup term is the prior-weighted variance of the subgroup means,

$$V_{\mathrm{cross}}(\ell) \triangleq \sum_{\mathcal{S} \in \mathcal{P}_\ell(\mathcal{X})} \gamma_{\mathcal{S}}\, (\mu_{\mathcal{S}} - \mu)^2 \qquad \text{with} \qquad \mu \triangleq \sum_{\mathcal{S} \in \mathcal{P}_\ell(\mathcal{X})} \gamma_{\mathcal{S}}\, \mu_{\mathcal{S}}.$$

By the law of total variance, these quantities satisfy $\mathbb{V}(Y) = V_{\mathrm{within}}(\ell) + V_{\mathrm{cross}}(\ell)$ for every level ℓ, up to finite-sample estimation error. Thus, the two terms quantify how refinement shifts variation from residual within-subgroup heterogeneity to explicit cross-subgroup heterogeneity.

F.6 Self-consistency evaluation on the binned ACS income setting

For the self-consistency analysis in Table 2a, we use the binned ACS income prediction task from the running example in Figure 14. The model is asked to estimate the income distribution over the following four bins:

Bin 1: $[-11.5\mathrm{k}, 1)$ USD

Bin 2: $[1, 25\mathrm{k})$ USD

Bin 3: $[25\mathrm{k}, 60\mathrm{k})$ USD

Bin 4: $[60\mathrm{k}, 1849\mathrm{k}]$ USD

These bins cover the full income range observed in the ACS sample: no individual reports an income below $-11.5\mathrm{k}$ USD or above $1849\mathrm{k}$ USD. The results in Table 2a report the two benchmark scores introduced in Section 5. The first is the split consistency score $\mathrm{SC}_\varepsilon(\mathcal{A})$ from Definition 2. For every admissible pair of a conditioning event and a next split attribute in $\mathcal{C}_{\mathrm{SC}}(\mathcal{A})$, we compare the model's direct estimate at the conditioning event to the prior-weighted aggregate of its estimates at the two refinements induced by the split attribute. For the attribute set $\mathcal{A}$ consisting of age and employment status, this yields six node-level checks. The second is the order consistency score $\mathrm{OC}_\varepsilon(\mathcal{A})$ from Definition 4. For each of the four two-attribute conditioning events in $\mathcal{C}_{\mathrm{OC}}(\mathcal{A})$, we elicit direct estimates under both orderings of the attribute constraints and check whether the two predictions agree. Since the target is an ordered categorical distribution, discrepancies are measured using the normalized Wasserstein distance $\overline{\mathcal{W}}_1$. Each score reports the fraction of checks whose discrepancy is less than the tolerance $\varepsilon = 0.02$.

F.7 Self-consistency evaluation on the thresholded ACS income setting

For the model comparison in Figure 4, we use the thresholded ACS income prediction task with threshold $\tau = 40\mathrm{k}$ USD. The setting mirrors the binned income experiment in Appendix F.6. We use the same conditioning tree over the attribute set $\mathcal{A}$ consisting of age and employment status. The only

change is the target variable, which is now the binary indicator of whether an individual's income exceeds τ.

For each model, we compute the split consistency score $\mathrm{SC}_\varepsilon(\mathcal{A})$ and the order consistency score $\mathrm{OC}_\varepsilon(\mathcal{A})$ at tolerance $\varepsilon = 0.02$. Since the target is binary, the normalized Wasserstein distance $\overline{\mathcal{W}}_1$ reduces to the absolute difference between the two estimated probabilities of exceeding the threshold. As a proxy for general model capability, we report the Artificial Analysis Intelligence (AAI) Index [Artificial Analysis, 2026]. Models are grouped by family and ordered within each family by their AAI Index.

F.8 Self-consistency evaluation on the ACS commute time setting

For the self-consistency analysis in Table 2b, we use the binned ACS commute time prediction task. The setting mirrors the binned income experiment in Appendix F.6. We use the same conditioning tree over the attribute set $\mathcal{A}$ consisting of age and employment status, and the same subgroup priors. The only change is the target variable, which is now the commute time in minutes. The model is asked to estimate the commute time distribution over the following four bins:

Bin 1: $[0, 1)$ minute

Bin 2: $[1, 15)$ minutes

Bin 3: $[15, 30)$ minutes

Bin 4: $[30, 195]$ minutes

These bins cover the full range of commute times observed in the ACS sample. No individual reports a commute time longer than 195 minutes. We compute the split consistency score $\mathrm{SC}_\varepsilon(\mathcal{A})$ and the order consistency score $\mathrm{OC}_\varepsilon(\mathcal{A})$ at tolerance $\varepsilon = 0.02$, exactly as in Appendix F.6. Since the target is an ordered categorical distribution, discrepancies are measured using the normalized Wasserstein distance $\overline{\mathcal{W}}_1$.

F.9 Self-consistency evaluation on WVS opinion distributions

We evaluate self-consistency on opinion distributions derived from the World Values Survey (WVS). The experiment reported in Table 3 uses data from Canada and Indonesia. For both countries, the evaluation is based on the attribute set $\mathcal{A}$ consisting of the following two splits:

Level 1: Age is split into "16 – 44 years old" and "45 – 103 years old".

Level 2: Income is split into "low" and "high".

The interpretation of "low" and "high" in the income split is left to the model. This is sufficient for our self-consistency checks, provided that the two options are interpreted as complementary categories that form a valid partition of the relevant population. For each country and each of the five selected WVS questions listed in Table 8, we elicit categorical response distributions at every conditioning event induced by $\mathcal{A}$. The corresponding prompt template is given in Appendix H.5. Each question then defines a distinct prediction problem, on which we compute the split consistency score $\mathrm{SC}_\varepsilon(\mathcal{A})$ and the order consistency score $\mathrm{OC}_\varepsilon(\mathcal{A})$ at tolerance $\varepsilon = 0.02$. For the split consistency score, we evaluate all admissible pairs of a conditioning event and a next split attribute in $\mathcal{C}_{\mathrm{SC}}(\mathcal{A})$, comparing the direct estimate at the conditioning event to the prior-weighted aggregate of its two refinements. For the order consistency score, we elicit direct estimates under both orderings of the attribute constraints for each of the four two-attribute conditioning events in $\mathcal{C}_{\mathrm{OC}}(\mathcal{A})$ and check whether the two predictions agree. Discrepancies are measured using the normalized Wasserstein distance $\overline{\mathcal{W}}_1$, and each score reports the fraction of checks whose discrepancy is less than $\varepsilon = 0.02$.

F.10 Self-consistency evaluation of the tennis forecasting example

Consider a forecasting task in which a model is asked to estimate the probability that Roger Federer wins a match against Rafael Nadal, conditional on contextual information available before or during the match. All estimates are elicited using the tennis forecasting prompt template in Appendix H.6. At the root node, the prompt provides only the matchup itself, corresponding to a blind or unconditional

forecast without additional information. We then gradually reveal pieces of contextual information that may become available before or during the match, such as the court surface. These conditions induce a binary conditioning tree analogous to the BCTs used in the income prediction setting, but the conditioning variables now describe a sporting event rather than a demographic subpopulation. Concretely, the attribute set $\mathcal{A}$ consists of two binary conditions: the court surface, distinguishing clay from non-clay courts, and the in-match event of whether Federer wins the first set. The former captures a pre-match contextual factor that is highly relevant to the Federer–Nadal rivalry, while the latter represents partial information that becomes available during the match. The same self-consistency checks from Section 5 can then be applied, with the model's estimated prior probabilities serving as aggregation weights.

To generate the results in Table 1a, we compute the split consistency score $\mathrm{SC}_\varepsilon(\mathcal{A})$ and the order consistency score $\mathrm{OC}_\varepsilon(\mathcal{A})$ at tolerance $\varepsilon = 0.02$. Since the target outcome is binary, discrepancies are measured by the absolute difference between the corresponding win probabilities. Each score reports the fraction of checks whose discrepancy is less than $\varepsilon = 0.02$.

F.11 Self-consistency evaluation of the fantasy example

As a second synthetic example, we consider a reasoning task in a fictional fantasy role-playing game. The model is asked to estimate the probability that the main character wins an encounter against an opponent using the prompt template in Appendix H.7. Unlike the survey setting, and unlike the tennis example, this task need not correspond to any real-world data-generating process. The game world is imaginary, and there is no external ground-truth distribution against which the model's estimates can be evaluated. This makes the example useful for highlighting a key feature of our self-consistency checks. They only require the model's own direct and conditional estimates, and therefore remain meaningful even when no reference distribution exists. At the root node, the prompt specifies only the basic combat situation. We then refine the scenario by revealing additional contextual information about the game world. The attribute set $\mathcal{A}$ consists of two binary conditions: the time of day, distinguishing day and night, and the wind conditions, distinguishing calm from windy combat environments.

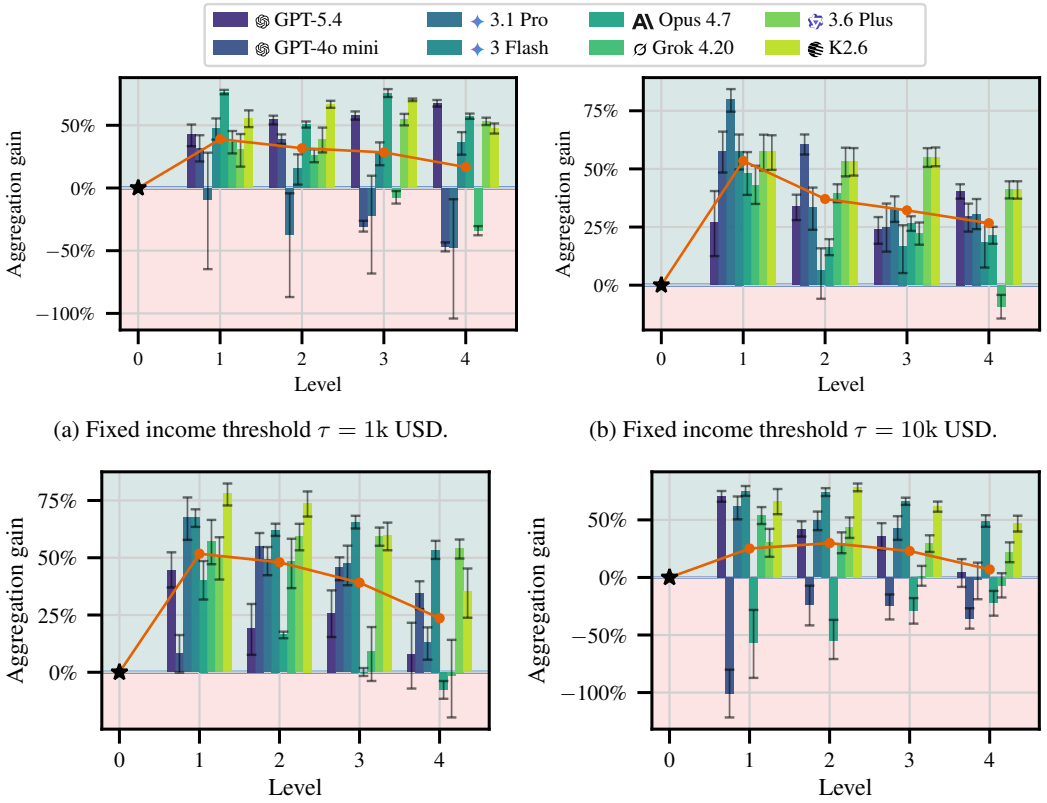
To generate the results in Table 1b, we proceed exactly as in the tennis example (Appendix F.10). We compute the split consistency score $\mathrm{SC}_\varepsilon(\mathcal{A})$ over the admissible checks in $\mathcal{C}_{\mathrm{SC}}(\mathcal{A})$ and the order consistency score $\mathrm{OC}_\varepsilon(\mathcal{A})$ over the four two-attribute conditioning events in $\mathcal{C}_{\mathrm{OC}}(\mathcal{A})$. Consistency requires the compared estimates to agree, even though the underlying event is fictional and no ground-truth win probability is available. Since the target outcome is binary, discrepancies are measured by the absolute difference between the corresponding win probabilities, and each score reports the fraction of checks whose discrepancy is less than $\varepsilon = 0.02$.

G Additional plots

This appendix collects larger plots with more detailed breakdowns.

G.1 Aggregation gain

We provide additional results on the aggregation gain to complement the main findings in Section 4. While Figure 2 highlights representative settings, Figure 16 reports the aggregation gain across a broader range of fixed income thresholds. These results show that the qualitative pattern from the main text is robust. Reconstructed aggregate estimates from subgroup conditionals are often better aligned with the ACS reference statistic than direct estimates, although the gains vary across thresholds, models, and tree levels.



(a) Fixed income threshold $\tau = 1$k USD.

(b) Fixed income threshold $\tau = 10$k USD.

(c) Fixed income threshold $\tau = 50$k USD.

(d) Fixed income threshold $\tau = 80$k USD.

Figure 16: **Aggregation gain across models and tree levels.** We report the aggregation gain $1 - \mathrm{AErr}(\ell)/\mathrm{AErr}(0)$ with $\mathrm{AErr}(\ell)$ defined in (3). Positive values indicate that reconstructing the population-level estimate from subgroup estimates improves alignment relative to direct prompting, while negative values indicate worse alignment. Each bar corresponds to a different model. Error bars show 90 % confidence intervals computed by bootstrapping with 1000 samples. The orange curve shows the average aggregation gain.

G.2 Reconstructed aggregate distributions across tree levels

Figure 17 visualizes the population-level ACS income distributions reconstructed from each tree level using LLM-estimated subgroup priors.

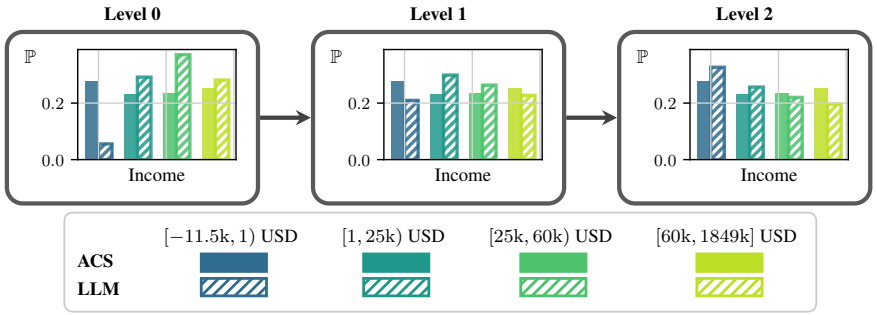


Figure 17: **Reconstructed aggregate income distributions across tree levels.** For each level of the binary conditioning tree in Figure 14, we reconstruct the population-level ACS income distribution by aggregating the subgroup distributions with subgroup priors. Solid bars show the same survey-based ground-truth distribution of the base population in all panels, while hatched bars show the corresponding LLM-induced reconstructed aggregate. The left panel corresponds to the direct estimate at the root, whereas the middle and right panels show aggregates reconstructed from increasingly fine-grained conditional estimates. The comparison illustrates that the reconstructed aggregate becomes better aligned with the ground-truth distribution as we move down the tree.

H Prompt templates

This appendix reports the exact prompt templates used to elicit probability estimates from language models.

H.1 Prior elicitation

Subgroup priors

```
You are given a mutually exclusive and exhaustive partition of the U.S.
population into groups defined by sociodemographic attributes.

For each group below, estimate the probability that a randomly sampled person
from the U.S. population belongs to that group.

{groups}

Important:
- Treat the listed groups as the full population partition.
- Assign probability mass across all groups.
- The returned probabilities must sum to 1.
- Use your best demographic knowledge of the U.S. population.

Return only a valid JSON object of the following form:
{
  "Group 1": 0.35,
  "Group 2": 0.65,
  ...
}

Strict output rules:
- Do not include any text outside the JSON object.
- Do not include explanations or uncertainty intervals.
- Each probability must be a single number in [0, 1].
- The JSON object must contain exactly one entry per group.
- Use the group names exactly as given: Group 1, Group 2.
```

The `groups` placeholder is replaced by a list of groups, each specified by a structured list of sociodemographic attributes that define the corresponding subgroup in the partition. For example:

```
Group 1
- sex: female

Group 2
- sex: male
```

H.2 Sociodemographic prompting

Thresholded income task

```
You are given a brief sociodemographic profile of an individual. Based on the
information provided and your general knowledge of income distributions in the
United States, estimate the probability of the specified outcome. The data
provided is sufficient to make an approximate probabilistic estimate.

Individual attributes:
{attributes}

Question:
What is the estimated probability that this person's annual income exceeds
{income_threshold} USD?
```

```
Output requirements (strict):
- Return a single real number between 0 and 1 (inclusive).
- Format the answer exactly as: [[p]] where p is your probability estimate (for
example: [[0.5]]).
```

In the sociodemographic formulation, the `attributes` placeholder is replaced by a structured list of demographic attributes describing the target individual. For example:

```
Individual attributes:
- Sex: Female
- Employment status: Civilian employed, at work
- Age: 24 years old
```

This formulation describes the conditioning information in the third person and avoids asking the model to adopt a role-playing perspective.

H.3 Persona prompting

Thresholded income task

```
Please answer the following question from the perspective of the described
person. Stay consistent with the profile, but do not mention that you are
role-playing unless asked.

Persona:
{persona}

Question:
What is the estimated probability that this person's annual income exceeds
{income_threshold} USD?

Output requirements (strict):
- Return a single real number between 0 and 1 (inclusive).
- Format the answer exactly as: [[p]] where p is your probability estimate (for
example: [[0.5]]).
```

In the persona formulation, the `persona` placeholder is replaced by a free-text description that conveys the same demographic information in a first-person role-playing style. For example, the profile may read: "You are a person who lives in the United States. Your sex is Female."

H.4 Partition-then-aggregate prompting

Thresholded income task

```
Based on your general knowledge of income distributions in the United States,
estimate the probability of the specified outcome. Please make an approximate
probabilistic estimate.

Question:
What is the estimated probability that a randomly selected person's annual
income exceeds {income_threshold} USD?

Before giving your final answer, reason from micro to macro:
1. Identify several relevant subpopulations within the U.S. population that may
differ in income.
2. For each subpopulation, consider both its approximate prevalence in the U.S.
population and its likely probability of exceeding the income threshold.
3. Combine these subgroup-level considerations into a single aggregate estimate
for the U.S. population.

Output requirements (strict):
- Return a single real number between 0 and 1 (inclusive).
```

```
- Format the answer exactly as: [[p]] where p is your probability estimate (for
example: [[0.5]]).
```

This prompt asks the model to estimate a population-level income probability by explicitly reasoning through relevant U.S. subpopulations, their prevalence, and their subgroup-specific probabilities. Unlike our tree-based aggregation prompts, the relevant subpopulations are not specified in advance but are instead chosen by the model.

H.5 Opinion distribution elicitation

Survey-response distribution prompt

```
You are estimating the distribution of survey responses for a demographic
subpopulation. The subpopulation is characterized by the following attributes:

{attribute_list}

Survey question:
{question}

Answer options:
{answer_option_list}

Task:
Estimate the probability distribution over the answer options for a randomly
selected individual from this subpopulation.

Strict output rules:
- Return only valid JSON.
- Use exactly the following keys: {answer_option_keys}.
- Each value must be a single number in [0, 1].
- The values must sum to 1.
- Use decimal probabilities, not percentages.
- Do not include explanations, comments, uncertainty intervals, or any text
outside the JSON object.

Return only a valid JSON object of the following form. The numbers below are
illustrative only and must be replaced by your estimated probabilities:
{example_output}
```

For the WVS opinion-distribution experiments, the `attribute_list` placeholder is replaced by the demographic attributes defining the target subpopulation, `question` is replaced by the survey question text, and `answer_option_list` lists the available response options. The model is instructed to return a complete categorical probability distribution over the answer options as a JSON object. The placeholders `answer_option_keys` and `example_output` ensure that the returned object has a fixed schema across models and prompts.

H.6 Tennis forecasting task

Tennis match forecasting prompt

```
You are estimating the outcome of a tennis match.

Match:
Roger Federer plays against Rafael Nadal.

The match context is characterized by the following attributes:
{attribute_list}

Forecasting question:
What is the probability that Roger Federer wins the match?
```

```
Task:
Estimate the probability that Roger Federer wins the match under the specified
context.

Strict output rules:
- Return a single real number between 0 and 1, inclusive.
- Use decimal probabilities, not percentages.
- Format the answer exactly as: [[p]], where p is your probability estimate.
- Do not include explanations, comments, uncertainty intervals, or any text
outside the required format.

Return only an answer of the following form. The number below is illustrative
only and must be replaced by your estimated probability: [[0.5]]
```

H.7 Fictional fantasy combat task

Fantasy combat forecasting prompt

```
You are estimating the outcome of a fantasy combat encounter.

A main character is engaged in combat against an enemy in an imaginary fantasy
setting. The world has a day-night cycle and varying weather conditions.

The current encounter context is characterized by the following attributes:
{attribute_list}

Forecasting question:
What is the probability that the main character wins the encounter?

Task:
Estimate the probability that the main character wins the encounter under the
specified context.

Strict output rules:
- Return a single real number between 0 and 1, inclusive.
- Use decimal probabilities, not percentages.
- Format the answer exactly as: [[p]], where p is your probability estimate.
- Do not include explanations, comments, uncertainty intervals, or any text
outside the required format.

Return only an answer of the following form. The number below is illustrative
only and must be replaced by your estimated probability: [[0.5]]
```

H.8 Conditional estimates from decision models

To elicit conditional estimates from the evaluated decision models, we use prompts comprising a question, a context, and a set of answer options, as illustrated below. The context attributes and their values are illustrative and may vary across the prompts presented to the models.

Question

```
Is this person's annual income greater than 40000 USD? Estimate the probability
under the specified context.
```

Context

```
The person is an individual in the United States with the following attributes:
- age: 40--60 years old
- class of worker: Currently employed or self-employed
- usual number of hours worked per week: 40--46 hours per week
```

Answer Options

```
{
  "A": "Yes, their annual income is greater than 40000 USD.",
  "B": "No, their annual income is at most 40000 USD."
}
```

H.9 Prior estimates from decision models

Prior estimates are elicited jointly for all nodes in a given tree level, with each node represented by an answer option describing its defining attributes.

Question

```
Which group does a randomly sampled person belong to? Estimate the probability
of each option under the specified context.
```

Context

```
A person is sampled uniformly at random from the U.S. population. Use your
background knowledge about the demographic composition of the United States.
```

Answer Options

```
{
  "Group 1": "The person has the following attributes: ...",
  "Group 2": "The person has the following attributes: ..."
}
```

I ACS attributes

Table 7 summarizes the ACS attributes used to define the conditioning variables in our experiments.

Attribute	Description	Values
AGEP	age	0 – 96 years old
COW	class of worker	1: "Working for a for-profit private company or organization" 2: "Working for a non-profit organization" 3: "Working for the local government" 4: "Working for the state government" 5: "Working for the federal government" 6: "Owner of non-incorporated business, professional practice, or farm" 7: "Owner of incorporated business, professional practice, or farm" 8: "Working without pay in a for-profit family business or farm" 9: "Unemployed and last worked 5 years ago or earlier or never worked"
ESR	employment status	1: "Civilian employed, at work" 2: "Civilian employed, with a job but not at work" 3: "Unemployed" 4: "Armed forces, at work" 5: "Armed forces, with a job but not at work" 6: "Not in labor force"
MAR	marital status	1: "Married" 2: "Widowed" 3: "Divorced" 4: "Separated" 5: "Never married"
OCCP	occupation	list of 530 occupations
WKHP	usual number of hours worked per week	1 – 99 hours

Table 7: **ACS attributes used for demographic conditioning.** We list the ACS variables used to construct the conditioning trees in our experiments, together with their descriptions and possible values. Integer-valued variables are reported by their observed range, while finite categorical variables are reported using their ACS response codes. For the demographic splits, we augment each value list with an n/a class to represent missing values in the ACS survey data.

J WVS questions

For completeness, Table 8 reports the World Values Survey questions used in our experiments, including their original WVS identifiers, question wording, and ordinal answer scales.

ID	WVS-ID	Question	Answer options
Q_a	Q162	Now, I would like to read some statements and ask how much you agree or disagree with each of these statements. For these questions, a 1 means that you "completely disagree" and a 10 means that you "completely agree": It is not important for me to know about science in my daily life	[1: Completely disagree, 2, 3, 4, 5, 6, 7, 8, 9, 10: Completely agree]
Q_b	Q177	Please tell me for each of the following statements whether you think it can always be justified, never be justified, or something in between, using this card. - Claiming government benefits to which you are not entitled	[1: Never justifiable, 2, 3, 4, 5, 6, 7, 8, 9, 10: Always justifiable]
Q_c	Q186	Please tell me for each of the following statements whether you think it can always be justified, never be justified, or something in between, using this card. Sex before marriage	[1: Never justifiable, 2, 3, 4, 5, 6, 7, 8, 9, 10: Always justifiable]
Q_d	Q245	Please tell me for each of the following things how essential you think it is as a characteristic of democracy. Use this scale where 1 means "not at all an essential characteristic of democracy" and 10 means it definitely is "an essential characteristic of democracy." - The army takes over when government is incompetent	[1: Not an essential characteristic of democracy, 2, 3, 4, 5, 6, 7, 8, 9, 10: An essential characteristic of democracy]
Q_e	Q252	How satisfied are you with how the political system is functioning in your country these days? On the scale from 1 to 10 below, 1 means not satisfied at all and 10 means completely satisfied.	[1: Not satisfied at all, 2, 3, 4, 5, 6, 7, 8, 9, 10: Completely satisfied]

Table 8: **Selected World Values Survey questions.** We report the five WVS questions used in our experiments, together with their original WVS identifiers, question text, and ordinal answer scales.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims are empirically justified and thoroughly discussed in the main paper and appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Throughout the paper we discuss the limitations of our work and further outline these in the discussion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All assumptions are stated explicitly, and complete proofs of all numbered results are given in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the codebase and reference to the survey data to reproduce all results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

Justification: We have open-sourced the code and included references to the required data.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experimental details are provided in the supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars are reported for the main empirical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This work primarily relies on LLM inference via publicly available inference providers. The required compute resources are therefore minimal.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: We do not expect any direct negative societal impact of our work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credit all the data we use and properly respect terms of use and licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.