CPUQ: Categorical Perplexity based Uncertainty Quantification with Language Models

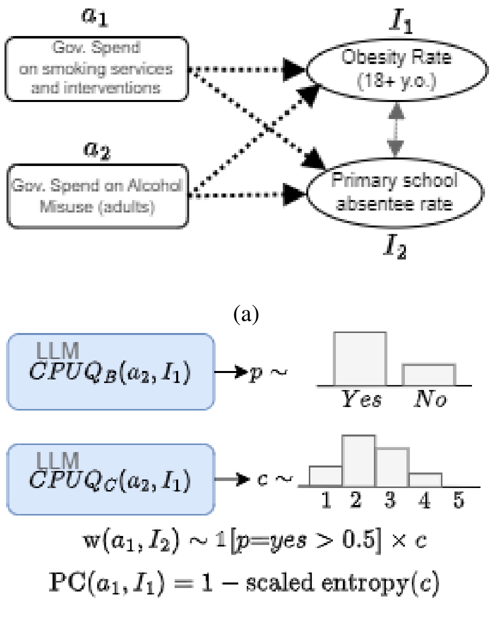
Anonymous ACL submission

Abstract

Agent-based modelling (ABM) for Economic Allocation (EA) analyzes interactions between economic agents and indicators over time, aiding policymakers and decision analysts in scenario analysis for complex systems like government spending or financial market contagion. These EA-ABMs, when graphed, often have limited datasets (timesteps) but a large number of nodes (agents or indicators) and edges (relationships), which hinders statistical network estimation methods. Additionally, statistical relationship estimation methods lack interpretability for non-technical users. To address these issues, we introduce the CPUQ framework, compatible with any Language Model (LM) and utilizing a LM's reasoning to generate predictive hurdle distributions that quantify relationship strength between agents/indicators, coupled with textual explanations for each prediction to enhance interpretability for non-technical audiences. CPUQ also includes a novel post-hoc calibration approach for network estimation. Evaluation on a real EA dataset demonstrates CPUQ's alignment with expert opinions and its superior forecasting capability over existing statistical and LM methods in assessing relationships in EA-ABMs.

1 Introduction

EA-ABM Economic Allocation (EA) Agent-based modelling (ABM) is utilized by policymakers and decision analysts to simulate interactions among economic agents and indicators, aiding in scenario analysis and forecasting in complex systems like government spending and financial market stress tests. These systems, characterized by a vast number of agents, indicators, and interactions, often face data limitations. Consequently, the effectiveness of EA-ABM is restricted by challenges inherent in statistical network estimation methods (§ 6) in settings with limited time-series data but numerous nodes and edges.


a_1
I_1
Gov. Spend on smoking services and interventions
Obesity Rate (18+ y.o.)
a_2
Gov. Spend on Alcohol Misuse (adults)
Primary school absentee rate
I_2
LLM
$CPUQ_B(a_2, I_1)$
$p \sim$
Yes
No
LLM
$CPUQ_C(a_2, I_1)$
$c \sim$
1 2 3 4 5
$\mathrm{w}(a_1, I_2) \sim \mathbb{1}[p{=}yes > 0.5] \times c$
$\mathrm{PC}(a_1, I_1) = 1 - \text{scaled entropy}(c)$

(a)

(b)

Figure 1: Figure a) shows an **Economic Allocation Network** representing the interactions between economic agents ($a_{\{1,2\}}$) and indicators ($I_{\{1,2\}}$) in a system modelling the effect of Government Spending on socio-health indicators. Figure b) shows the LM agnostic CPUQ framework which performs network estimation, determining predictive hurdle distributions $w(\cdot)$ for edge weights. CPUQ$_{B,C}$ produces a Bernoulli or Categorical Distribution determining edge existence and conditional edge weight respectively. We perform Network pruning by thresholding our measure of predictive certainty $PC(\cdot)$, as opposed to the magnitude of the weight.

CPUQ Our approach, Categorical Perplexity based Uncertainty Quantification (CPUQ), estimates edge weights in Text Attribute Graphs (TAG) (Figure 1, Table 1). It outputs a hurdle mixture distribution, combining a Bernoulli distribution for modeling the possibility of no edge, and a Categorical distribution for the edge weight if present. Relative to statistical network estimation methods

CPUQ Stage	Prompt Template
Question & Reason	Write a thorough, detailed and conclusive four-sentence answer to the following question. To what extent, if any, is the level of {indicator 1} influential to the state of {indicator 2}?
CPUQ_B Prompt	Write only the number of the category that fits the following statement. "Statement: [Model's response to Q&R Stage]" Categories: 1) The level of {indicator 1} is {effect type} influential to the state of {indicator2}. 2) The level of {indicator 1} is not {effect type} influential to the state of {indicator2}.
CPUQ_C Prompt	On a scale of 1 to 5, how strong is the influence of changes in {indicator 1} on changes in {indicator 2}?

Table 1: **Example Prompt Templates:** Examples of prompts for predicting indicator to indicator relationships in our Economic Allocation experiments. The {indicator} placeholders represent textual representations of indicators. {effect_type} can be 'direct', 'indirect', or blank. $\text{CPUQ}_{B/C}$ denote the CPUQ methods yielding Bernoulli and Hurdle Categorical Distributions based on model perplexity. The sequential prompts (Prompt 1-3) illustrate the conversational context approach, used by the CPUQ method.

the use of text attributes better reflects causation modelling. CPUQ also provides interpretable textual explanations, promotes network sparsity with predictive hurdle distributions and allows for network pruning based on predictive certainty.

We validate CPUQ on a UK regional government's Economic Allocation system, aiming to allocate budgets effectively over a 9-year horizon. We address conceptual and practical challenges with uncertainty quantification in language models and check for biases by comparing edge distribution predictions to existing methods. CPUQ's advantages include alignment with expert-annotated datasets, interpretability, cost-effectiveness, and robust uncertainty quantification.

The essence of our contributions lies in:

- Develop CPUQ, a LM agnostic framework using categorical question prompts to output predictive hurdle categorical distributions coupled with interpretable textual explanations.
- Motivate CPUQ to solve conflation of semantics and syntax when performing quantitative question answering.
- Show that CPUQ improves EA-ABM forecasting performance relative to existing statistical and LM based methods while attaining strong alignment with expert annotations.
- Validate our proposed method of network pruning with thresholds on predictive certainty.

2 Uncertainty Quantification Challenges

Previous works have experimented with using various forms of sampling based approaches to Uncertainty Quantification which we discuss below.

Prompt Variation Methods: Prompt variation uses semantically similar prompts to induce stochasticity in output. A large body of works (Arora et al., 2022; Wei et al., 2022) have demonstrated strong performance increases on QA tasks by designing methods that search for an optimal prompt within a semantically similar set of prompts. This line of research suggests prompt variation does not test a model's predictive uncertainty but mostly the quality of the prompts or language model. Further supporting this, Jiang et al. (2021) showed that prompt specification becomes less important as the foundational models become better calibrated.

Sequence Perplexity Based Measures: Text sequence probability, s, is derived from the product of conditional probabilities of new tokens given past tokens, leading to a log-probability $\log p(\mathbf{s} \mid x) = \sum_i \log p\left(\mathbf{s}_i \mid \mathbf{s}_{<i}\right)$ where $\mathbf{s}_i$ is the i'th output token and $\mathbf{s}_{<i}$ denotes the set of previous tokens. Previous works (Jiang et al., 2021; Kuhn et al., 2023) utilized predictive entropy $H(s \mid x) = -\int p(s \mid x) \ln p(s \mid x) dx$ as an uncertainty measure, while others (Malinin and Gales, 2018; Murray and Chiang, 2018) employed the average log-probability $\frac{1}{N} \sum_{i=1}^{N} \log p\left(s_i \mid \mathbf{s}_{<i}\right)$.

Prior studies (Kuhn et al., 2023) noted the lack of theoretical backing for these methods. We build on this by introducing a formal condition applicable when tokenized sequences s exceed length 1.

When considering sequences over length 1, the conditional probability $p\left(\mathbf{s}_i \mid \textbf{concat}(x, \mathbf{s}_{<i})\right)$ has theoretically (Mann and Thompson, 1987) and practically (Adewoyin et al., 2022; Banarescu et al., 2013) been decomposed into composite distributions over syntax and semantics, where syntax is the arrangement of words and phrases to create well-formed text and semantics is the underlying meaning of the text. With this understanding, sequence perplexity can be seen to incorrectly measure uncertainty over syntax and semantics as opposed to solely uncertainty over semantics.

We affirm that prior research (Murray and Chiang, 2018) illustrated instances of this issue, like ’label bias’ impacting response lengths, and others (Jiang et al., 2021; Kuhn et al., 2023) demonstrated biases towards varying expression styles with ‘semantic equivalence’.

Overcoming Stylistic Bias When language model responses are limited to one-token sequences, stylistic syntax has minimal impact on output. Typically, in Yes/No questions, the response (‘Yes.’ or ‘No.’) is unaffected by syntactic style. This concept is illustrated by decomposing the output sequence s conditional probability, given prompt x, $p(\mathbf{s} \mid x) = \sum_i \log p\left(s_i \mid \textbf{concat}(x, \mathbf{s}_{<i})\right)$, into a joint probability with a latent semantic meaning $m \in M$ as shown below:

$$\log p(\mathbf{s} \mid x) = \sum_{m \in M} \log p(\mathbf{s}, m \mid x) \tag{1}$$

$$\begin{aligned} = \sum_m \sum_i \log p\left(s_i \mid \textbf{concat}(x, \mathbf{s}_{<i}), m\right) \\ + \log p(m \mid x) \end{aligned} \tag{2}$$

$$= \sum_m \log p\left(s_0 \mid x, m\right) + \log p(m \mid x) \tag{3}$$

$$\approx \log p\left(s_0 = s^* \mid x, m\right) + \log p(m \mid x) \tag{4}$$

Equation 2 explains the surface realization s as a two-step process: first modeling semantic meaning m, then conditionally modeling s over a bivariate distribution of prompt x and latent semantic meaning m. Equation 3 simplifies this due to one-token response limits. Equation 4 then restricts our prompt x to a set of Prompts X', such that for each semantic meaning m our language model's output range is effectively constrained to a single token $s^m, p(\mathbf{s}_0 = s^m \mid x, m)$.

As $p(\mathbf{s}_0 = s^m \mid x, m)$ nears 1 for $m \in M$, output variability $p(\mathbf{s} \mid x)$ mainly stems from $\log p(m \mid x)$, highlighting that in single-token responses to certain prompts $x \in X'$, uncertainty is largely due to latent semantics, not stylistic response variations.

We address this with categorical question style prompts.

3 Categorical Perplexity based Uncertainty Quantification

Section 2 introduced Categorical Prompts for Uncertainty Quantification (CPUQ) as a more effective alternative to sequence sampling methods, focusing on uncertainty over distinct semantic rather than syntactic outputs.

We remind the reader that our downstream task is network estimation for the systems modelled by EA-ABMs, for which we determine a probabilistic distribution over edge existence and edge weight. Table 1 provides prompt templates and Figure 1 provides an illustration for the following four steps:

1. Prompting the LM for a Reasoned Answer
2. Edge Existence Prediction CPUQ$_B$
3. Edge Weight Prediction CPUQ$_C$
4. Entropy based Network Pruning

1. Prompt the LM for a Reasoned Answer Building on previous research (Wei et al., 2022; Zhang et al., 2022; Wang et al., 2023), which showed improved QA performance when the deductive process is broken into intermediary steps, we tested CPUQ with prompts for intermediary explanations before the final answer, as shown in Figure 1.

2. Edge Existence Prediction CPUQ$_B$ For edge existence, we create a categorical question style prompt that requires the model's response to be one number corresponding to the correct category number. We then use the perplexity over a one-token output space $('1','2')$ to create a Bernoulli distribution, which models the probability of the edge existing or not existing.

3. Edge Weight Prediction CPUQ$_C$ Upon reaching a threshold 0.5, a categorical prompt asks for a single-digit number reflecting relationship strength between two agents / indicators. The categorical mean weight is calculated using normalized likelihoods over the output space as $p_{norm}(s^i \mid$

$x) = \frac{f(s^i|x)}{\sum_{j \in J} f(s^j|x)}$ and $\mu(s) = \sum_{i=1}^{5} s^i \cdot p_{norm}(s^i \mid x)$.

4. Entropy based Network Pruning A prevalent post-hoc calibration method is edge pruning through placing a minimum threshold on edge weight. In contrast, we propose edge pruning on the entropy of our predictive distributions, which achieves pruning on the basis of prediction certainty rather than the magnitude of the prediction.

We threshold Predictive Certainty $PC = 1 - H(p)$, where $H(p)$ is a base-scaled entropy measure. PC inverts values to indicate maximum certainty/uncertainty for 1/0. For the CPUQ$_B$, the scaled entropy $H_B(p)$ is $H_B(p) = -(p\log_2 p + (1-p)\log_2(1-p))$. For CPUQ$_C$, it's $H_C(p) = 1 + \sum_{i=1}^{5} p_i \log_5 p_i$, where p_i is the i-th value's probability.

3.1 Post-hoc Calibration Methods

Unbiasing Categorical Label Order We observed stylistic biases in initial CPUQ$_B$ experiments, favoring either the first or second categorical response, e.g. consistently inflating probability of 1. To counteract this, we implemented a method where the same question is posed twice with reversed categorical response order, and then averaged the two response distributions. We noticed that the significance of this form of label bias was negatively related to the strength of the underlying language model.

Finetuning Following prior research (Jiang et al., 2021), we finetuned models on domain-specific knowledge. Models up to 17bn parameters were fine-tuned due to hardware constraints. We used an instruction dataset and a Social Policy-focused text dataset, in equal proportions. This enhances the model's Social Policy expertise while maintaining instruction-following proficiency, ensuring the conditional distribution $p(s \mid x)$ favors relevant response tokens over text continuation. Details on these datasets are in Appendices D.1 & D.2.

Preventing Hallucination To reduce hallucination in LM responses during Question & Reason (Q&R), we designed prompts that implicitly limited response length to 4 sentences (Table 1), as current leading LMs typically don't hallucinate at this length. This strikes a balance between information retrieval and hallucination prevention. CPUQ, adaptable to any language model, should improve alongside language model advancements, potentially allowing for longer responses without increased hallucination risk.

Model	Prompt Style	F1	Prec.	Rec.
GPT3.5	verb_closed	0.795	0.722	0.883
GPT3.5	verb_open	0.830	0.779	0.888
30bn	verb_closed	0.767	0.715	0.826
30bn	verb_open	0.778	0.681	0.908
30bn	CPUQ$_B$ closed	0.698	0.757	0.647
30bn	CPUQ$_B$ Q&R	0.760	0.644	0.928

Table 2: **Expert Annotation Alignment:** This study compares the effectiveness of prompting methods in predicting the impact of local government budget items on socio-economic indicators. It contrasts deterministic Yes/No answers from verb_closed and verb_open strategies with CPUQ's probabilistic outputs. Refer to Table 1 for Prompt Styles examples. We use GPT3.5 and the 30bn llama model. Q&R signifies Question & Reason. CPUQ$_C$ shows competitive performance with verbalize and notably higher recall.

4 Validation: Alignment To Expert Annotation

We validate the degree to which CPUQ$_B$ predictions align with a dataset produced by the UK government which links government spending on broad budget items to the specific socio-economic indicators they affect.

Data We use a labelled network dataset from open-access UK government resources. It links spending on 15 broad budget items to 258 socio-economic indicators (Figure 1a). Details on dataset construction are in Appendix F. We supplement the dataset with negative samples, i.e., pairs of (budget item, indicator) with no relationship.

Model Our experiments use the llama language models with 7bn, 13bn, and 30bn parameters.

Baselines We compare our method to two baseline approaches (verb_open) and (verb_closed) based on methods from previous studies (Tian et al., 2023; Zhou et al., 2023; Lin et al., 2022), which simply prompt the model to verbalize its answer with an open-ended or close-ended response. We also compare to gpt3.5-turbo, providing insight into the effect of foundational model strength. Unlike the open-source LMs used for CPUQ, gpt3.5-turbo requires a paid API, and does not reveal log-probabilities preventing the use of CPUQ.

Results In this binary classification task, we present F1, Precision, and Recall scores (Table 2). All CPUQ methods perform competitively with the verbalize approaches which do not produce probabilistic outputs. The CPUQ Question & Reason outperforms the CPUQ Closed Ended Question, highlighting the benefit of intermediary reasoning. GPT3.5 shows the highest performance, emphasizing foundational model strength.

4.1 Ablation Experiments

We investigate the sensitivity of our approach to model size and our post-hoc calibration methods. For ablation, we include the Expected Calibration Error (ECE) metric (Guo et al., 2017). It quantifies the calibration quality of probabilistic predictions by computing a weighted average of the differences between observed accuracy and the predicted confidences across distinct intervals.

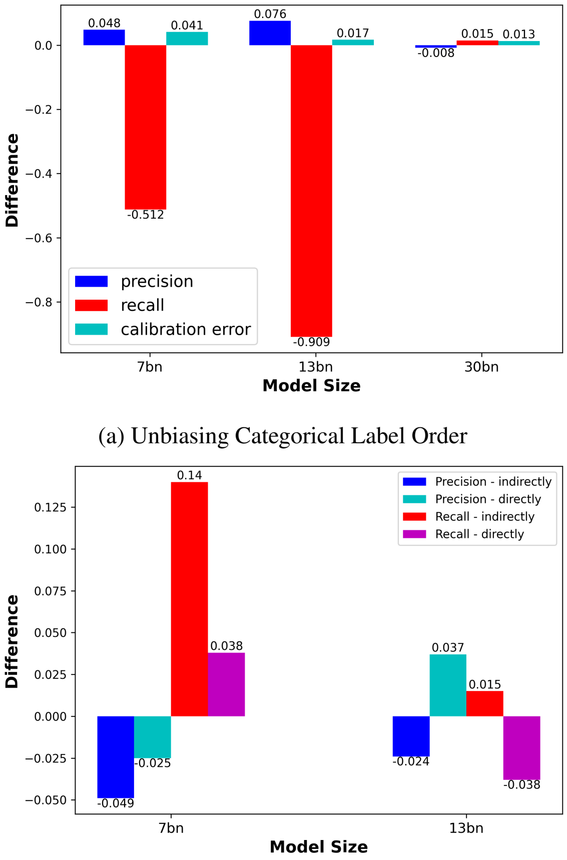
When mitigating stylistic bias in the CPUQ$_B$ method's label order, we observed significant recall degradation in foundational models smaller than 13bn parameters, while the 30bn model showed modest improvements in recall and Expected Calibration Error. Smaller models struggle with unconventional responses orderings such as 1) Negative Response 2) Affirmative Response.

For the 7bn and 13bn models, introducing 'indirectly' in prompts decreased precision, indicating these models consider broader relationships compared to expert annotators. However, this also led to increased recall in both models, suggesting a greater likelihood of identifying potential relationships.

5 Evaluation: EA-ABM Forecasting

We compare the forecasting performance of an EA-ABM algorithm called Policy Priority Inference (PPI) when the underlying network is estimated using our CPUQ methods and other baseline methods. For each method/network, we train the PPI system on the first 5 years of data, then evaluate predictions for the level of the socio-economic/health indicators for over the next two years.

The PPI algorithm models interactions between government spending budget item on indicator (b2i) and the second order spillover effects of indicator to indicator (i2i) interactions (Figure 1a). In the PPI algorithm the b2i edges are binary, while the i2i edges are variable weights, appropriate for our CPUQ$_B$ and CPUQ$_C$ methodologies respectively. For a detailed explanation of the PPI algorithm please refer to Appendix B.



(a) Unbiasing Categorical Label Order

(b) Varied Effect Type

Figure 2: **Ablation Experiments:** These figures show predictive performance in classifying edge existence in a Textual Attribute Network for an EA dataset related to UK government spending and socio-economic indicators. Figure a) illustrates the impact on predictive scores of applying our unbiasing method for categorical Label order (detailed in Section 3). Figure b) compares performance variations when specifying the prompt templates' "effect type" as 'directly', 'indirectly' or non-specified. This prompt template is detailed in Table 1.

Data Our dataset spans 7 years of annual data on UK government spending for 32 fine-grained health-related government spending budget items, sourced from the Spend and Outcomes Tool (SPOT), and 258 socioeconomic-health indicators from various public entities, including Fingertips and the UK Office for National Statistics. The first 5 years constitute the training set, while the last two years serve as forecasting test set. This includes 8256 potential b2i edges and 66564 i2i edges for estimation. More detailed information is available in Appendix F, which also outlines the connection between broad government budget items and specific indicators.

Baseline Methods Each experimental result consists of methods for predicting both b2i and i2i edges independently. For determining the b2i edges, baseline methods include verbalize with close-ended questions as detailed in Table 1 and naive expert annotation (n.e.a.). The latter extends the expert annotation—which provides related pairs of broad budget items b_b and indicators i by assuming every fine-grained budget item b_f that's part of the broad budget item ($b_f \in b_b$) relates to all the indicators the broad budget item is noted to connect with: if $b_f {\in} b_b$, and $(b_b, i){\to}(b_f, i)$.

For determining i2i edges, baseline methods encompass zero (representing no spillover effects between indicators), verbalize as shown in Table 1, entropy of the CPUQ$_B$ output bernoulli distribution for all edges with a probability over 0.5 of existing, and the Concave penalized Coordinate Descent with reparameterization (CCDr) algorithm. CCDr estimates Bayesian network structures using penalized maximum likelihood estimation combined with coordinate descent optimization on reparameterized Gaussian likelihoods. By inducing convexity in the likelihood and applying sparsity-inducing MCP (Li et al., 2022) regularization, it efficiently learns graphs, especially in $p >> n$ scenarios. Details on the CCDr methodology can be found in Appendix C.

For the CPUQ and verbalize methods, we employ a model from a 30bn parameter set of the llama family, finetuned on our curated datasets as described in Appendices D.1 & D.2.

5.1 Results

For the set of experiments where the b2i methodology is fixed to naive expert annotation (n.a.e.) and i2i method varies, in Table 3 we observe that CPUQ$_C$ method achieves the lowest mse/mae score, performing competitively with the entropy approach.

For the set of experiments where we additionally predict the b2i edges, we immediately notice a degradation in performance of the verbalize method and CPUQ method, indicating relative difficulty in predicting b2i relative to i2i edges. We posit this is due to binary output space of the b2i edges meaning that mis-specification of an edge weight has a larger negative effect on performance. Within this category we notice the CPUQ approach outperform the verbalize approach.

b2i	i2i	mse	mae
n.e.a	zero	0.01208	0.04835
n.e.a	CCDr	0.01209	0.04832
n.e.a	entropy	0.01196	0.04822
n.e.a	verbalize	0.01200	0.04814
n.e.a	CPUQ$_C$	0.01195	0.04820
verbalize	verbalize	0.01211	0.04830
CPUQ$_B$	CPUQ$_C$	0.01202	0.04825

Table 3: **PPI Forecasting Performance:** Prompting methodologies are varied for prediction of binary budget item to indicator (b2i) and non-binary indicator to indicator (i2i) relationships. For b2i edges, methods include naive expert annotation (n.e.a) and verbalize. Float i2i methods include zero (no spillover), verbalize, entropy from CPUQ$_B$ with > 0.5 probability, the CCDr algorithm and CPUQ$_C$(Section 3). Results highlight the competitive performance of CPUQ$_C$ and the relative difficulty of labelling binary valued edges.

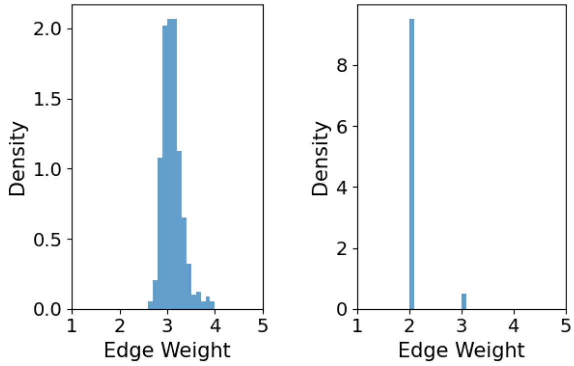
The second set of experiments focus on also predicting the binary b2i edges in the network as well as the non-binary i2i edges in the network. We notice that CPUQ outperforms the verbalize method.

5.2 Inspecting Edges Distribution

In Figure 3a we show the distribution of values for the predicted values for the i2i edges in our Economic Allocation network. The verbalize method exhibits an output limited to two values of 2.0 and 3.0. Conversely, we notice that the CPUQ$_C$ method produces a unimodal distribution centered around 3.0 with tails extending to 2.6 and 4.0.

5.3 Entropy Thresholding

In Figure 4 we evaluate the effect on forecasting performance when applying different thresholding strategies for the weights predicted by CPUQ$_C$, CCDr and Verbalize. For the network edge weights predicted by CPUQ$_C$, thresholding is applied to the scaled entropy of the predictive categorical distribution. For the network weights predicted by CCDr and Verbalize, thresholding is based on the absolute value of the point predictions for the weights. For each method we use fifteen linearly spaced thresholds over an appropriate range. For CPUQ$_C$, the fifteen ordered thresholds are the values at 0.05 intervals starting from 0.0 and ending at 0.7. For Verbalize, the fifteen ordered thresholds are values at 0.5 intervals starting from 0.0 and ending at 7.0.



(a) CPUQ$_C$ (b) Verbalize

Figure 3: **Distribution of Predicted Edge Weights**: We compare the distribution of non-zero predicted edge weights from our CPUQ$_C$ prompting strategy to the distribution of edges from verbalize strategy when using the same underlying language model. We notice the verbalize exhibits a limited distribution with values falling on the values of 2 and 3. Our CPUQ$_C$ approach values in the range of 2.6 and 4.0.

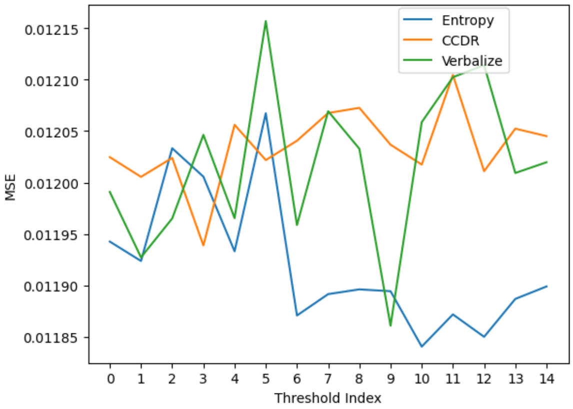


Figure 4: **Entropy Thresholding**: We evaluate the effect on forecasting performance when applying different thresholding strategies for the weights predicted by CPUQ$_C$, CCDr and Verbalize. For the network weights predicted by CPUQ$_C$, thresholding is applied to the scaled entropy of the predictive categorical distribution. For the network edge weights predicted by CCDr and Verbalize, thresholding is based on the absolute value of the point predictions for the weights. For each method we use fifteen linearly spaced thresholds over an appropriate range. We notice that our entropy thresholding method produces a convex optimisation path with respect to threshold value, while threshold tuning CCDr and Verbalize do not exhibit a clear minima. The CPUQ$_C$ outperforms the other methods at the 10th threshold which corresponds to a minimum entropy of 0.5.

With tuned thresholding, CPUQ$_C$ achieves an MSE of 0.1184, which is an improvement from the non tuned result of 0.1195 reported in Table 3. We notice that our entropy thresholding method produces a convex optimisation path with respect to threshold value, while threshold tuning CCDr and Verbalize do not exhibit a clear minima. We believe this is attributable to the fact that thresholding on the absolute value filters for point predicted weights with a high value, whereas thresholding on the predictive entropy filters for categorical predictions that exhibit high certainty over any value. Intuitively, this can be interpreted as absolute value thresholding using absolute weight value as a proxy for certainty, while entropy thresholding more correctly uses predictive entropy to model certainty. The ability to threshold on predictive entropy is a unique benefit of our CPUQ$_C$ approach that produces a categorical distribution.

6 Related Work

Network Estimation Approaches Statistical network estimation methods such as those based on Bayesian networks (BN) (Pearl, 1988; Massara et al., 2015; Aragam and Zhou, 2015) assume acyclic graphs and do not describe causal relationships, while Granger-causality networks based on Granger (1969); Kang et al. (2017) assume underlying linear relationships between variables (Castagneto-Gissey et al., 2014) and are inappropriate for test of predictability involving more than two variables. Further, these methods often require a sufficient observations-to-variables ratio, a common limitation for Economic Allocation Systems even with matrix factorization methods. This issue is addressed by Aragam and Zhou (2015) who propose Concave penalized Coordinate Descent with reparametrization (CCDr), a non-convex optimization approach that uses sparse regularization and block-cyclic coordinate descent.

Language Models (LM) present an alternative network estimataion method (Yamasaki et al., 2023; Bansal et al., 2019; Saxena et al., 2022) by building prompts from the textual attributes associated with nodes in the network (see Figure 1). Kıcıman et al. (2023) show that this LM approach can capture both associative and non complex causal relationships, in contrast to the BN methods which only model associative relationships. However, the ability to model complex causal relationships is limited, as Jin et al. (2023) found that performance suffers in out-of-distribution settings when variable names and textual expressions used in the prompts are dissimilar to those in the LM's training set.

Uncertainty Quantification Recent work have explored various approaches for quantifying uncertainty in predictions from large language models (LMs). Some methods have focused on eliciting and evaluating verbalized confidence scores produced by the LM itself (Tian et al., 2023; Zhou et al., 2023). Others have proposed using consistency among multiple candidate answers as a proxy for the model's uncertainty (Xiong et al., 2023; Ngu et al., 2023). While promising, these approaches don't rely on the standard measure of perplexity.

For example, Ngu et al. (2023) present domain-independent uncertainty measures based on the diversity of responses to a prompt, including entropy, Gini impurity, and centroid distance. They demonstrate these sample-based diversity measures correlate with failure probability without using perplexity. Similarly, Xiong et al. (2023) introduce consistency-based confidence scores by generating multiple candidate answers and assessing their consistency. They also propose hybrid methods combining consistency with verbalized scores. However, these methods require drawing multiple samples from already large Language Models leading to a large computational expense.

Other studies have focused on eliciting calibrated confidence estimates directly from language models fine-tuned with human feedback (Tian et al., 2023; Zhou et al., 2023; Lin et al., 2022). These methods produce probability scores or phrases representing the model's certainty, showing strong performance in calibration metrics. While promising, they rely less directly on perplexity itself. Both Lin et al. (2022) and Kadavath et al. (2022) also propose ways to predict models uncertainty by fine-tuning predictors on the embeddings of generating models. However, these approaches need task-specific labels, additional training, and are unreliable out-of-distribution (Kadavath et al., 2022).

Some prior work has addressed the important concern of grouping semantic similar terms when distributing probabilities over candidate answers. Jiang et al. (2021) study the case of one word answers by summing the probability over groups of synonyms, while Kuhn et al. (2023) extend this idea to phrases by grouping phrases which are deemed to have semantic equivalence. Although, both methods incur a large additional computational cost, as they require a secondary model which is used to evaluate similarity of different candidate answers and also utilize a sampling methodology. In contrast, CPUQ evaluates likelihood of categorical predictions from language models avoiding time-ineffeciency of sample-based techniques and inconsistencies of open-ended verbalized scoring.

7 Conclusion

We introduced CPUQ, a novel method for uncertainty quantification using Language Models. This method utilizes categorical-style questions to generate hurdle categorical distributions for edges in a Text Attribute Graph associated with Agent-based modelling for Economic Allocation. Validated against a UK dataset on government spending and socio-economic indicators, CPUQ aligns effectively with expert annotations, outperforms prominent alternative LM and statistical methods, and provides non-technical end-users with interpretable textual explanations.

Further Work Post-hoc methods to increase calibration of output distributions are important. Within this, we place specific relevance on methods that increase the variability of distributions produced when models are prompted for a scaled answer. Figure 3a showed that the CPUQ$_C$ method for predicting edges produces a set of distributions that may not span the entire range. Future work could also investigate the performance of methods that can extend this methodology to scenarios where the set of answers are not know ex-ante.

An alternative direction is to develop a network estimation approach that uses the textual attribute information of the nodes and as well as the limited time-series data available for them.

8 Ethics Statement

We acknowledge that our proposed model may be susceptible to having learnt harmful biases present in the pre-training and finetuning datasets. In and of itself, this has the potential to produce harmful suggestion for policymakers and decision makers. Therefore, we advocate for morally correct and responsible practices in the case of real-world applications. When creating our Social Policy Dataset, we aimed to choose a dataset with equal focus across different social policy topics and with restriction to articles published, in order to prevent the biased/harmful learning. Finally, as CPUQ is a framework wrapping any language model, it inherits the benefit of any safety measures implemented in the foundational model.

References

Rilwan Adewoyin, Ritabrata Dutta, and Yulan He. 2022. Rstgen: Imbuing fine-grained interpretable control into long-formtext generators. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1822–1835.

Bryon Aragam and Qing Zhou. 2015. Concave penalized estimation of sparse gaussian bayesian networks. *Journal of Machine Learning Research*, 16(69):2273–2328.

Simran Arora, Avanika Narayan, Mayee F. Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. 2022. Ask me anything: A simple strategy for prompting language models.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Trapit Bansal, Da-Cheng Juan, Sujith Ravi, and Andrew McCallum. 2019. A2N: Attending to neighbors for knowledge graph inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4387–4392, Florence, Italy. Association for Computational Linguistics.

G. Castagneto-Gissey, M. Chavez, and F. De Vico Fallani. 2014. Dynamic granger-causal networks of electricity spot prices: A novel approach to market integration. *Energy Economics*, 44:422–432.

Office for Health Improvement & Disparities (OHID). 2023. Public health profiles.

Office for National Statistics (ONS). 2023. Gross domestic product at market prices:implied deflator:sa.

C. W. J. Granger. 1969. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438.

Omar A. Guerrero and Gonzalo Castañeda. 2020. Policy priority inference: A computational framework to analyze the allocation of resources for the sustainable development goals. *Data amp; Policy*, 2:e17.

Omar A. Guerrero and Gonzalo Castañeda. 2021. Quantifying the coherence of development policy priorities. *Development Policy Review*, 39(2):155–180.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks.

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

Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona Diab, and Bernhard Schölkopf. 2023. Can large language models infer causation from correlation?

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Dongyeop Kang, Varun Gangal, Ang Lu, Zheng Chen, and Eduard Hovy. 2017. Detecting and explaining causes from text for a time series event. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2758–2767, Copenhagen, Denmark. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality.

Bowen Li, Suya Wu, Erin E. Tripp, Ali Pezeshki, and Vahid Tarokh. 2022. Minimax concave penalty regularized adaptive system identification.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023a. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023b. Wizardcoder: Empowering code large language models with evol-instruct.

Andrey Malinin and Mark Gales. 2018. Predictive uncertainty estimation via prior networks.

William C Mann and Sandra A Thompson. 1987. *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute Los Angeles.

Guido Previde Massara, T. Di Matteo, and Tomaso Aste. 2015. Network filtering for big data: Triangulated maximally filtered graph.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium. Association for Computational Linguistics.

Noel Ngu, Nathaniel Lee, and Paulo Shakarian. 2023. Diversity measures: Domain-independent proxies for failure in language model queries.

Office for Health Improvement & Disparities. 2023. Spend and outcomes tool.

Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-sequence knowledge graph completion and question answering.

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

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions.

Shohei Yamasaki, Yuya Sasaki, Panagiotis Karras, and Makoto Onizuka. 2023. Holistic prediction on a time-evolving attributed graph. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13676–13694, Toronto, Canada. Association for Computational Linguistics.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment.

A Economic Allocation Agent-based Modelling Systems

Agent-based Modelling (ABM) serves as an instrumental framework for depicting complex economic allocation games that involve interdependent agents. The delineation of the political economy game from the accompanying research can be broadened into three primary aspects: environment, agents, and dynamics.

Environment: The configuration presents a graph which elucidates the interdependencies among N agents, potentially characterized by general graph structures such as Erdős-Rényi or Barabási-Albert models. Every agent, denoted by i, encompasses a state variable S_i to manifest its prevailing state, which could span across either continuous or discrete realms. Furthermore, a global state S amalgamates the states of all agents.

Agents: In the context of agents, each i is driven to amplify a reward function $R_i(S)$, contingent on the global state, epitomizing the economic incentives intrinsic to every agent. An inherent limitation faced by the agents is the absence of comprehensive knowledge about the specific states or actions of their counterparts. Their observations remain confined to the local data discernible within their graph neighborhood.

Dynamics: With the progression of each time step t, every agent i institutes an action $A_i(t)$ rooted in their localized observations, culminating in the evolution of their individual state S_i. Owing to the intricate web of interdependencies embedded in the graph, modifications in the local state permeate, influencing the overarching global state S. Subsequently, the environment reciprocates by dispensing a reward $R_i(t)$ to each agent, in line with the recalibrated global state. The overarching goal for agents is to unravel behaviours that potentiate the maximization of long-term rewards through their actions. Potential learning algorithms might encompass model-free reinforcement learning, model-based planning, or heuristic adjustments analogous to the research.

This expansive framework offers the latitude to emulate diverse economic allocation scenarios within the ambit of multi-agent games. The intricate graph structure translates the dependencies, while the local observations of agents stand as proxies for the imperfect information. Meanwhile, the learned policies illuminate the underlying incentives and adaptations. In tandem, the platform facilitates a comparative study of different learning algorithms, focusing on global efficiency and equity outcomes, rendering it an ideal bedrock for delving deep into decentralized economic systems.

B Policy Priority Inference

In this section we provide a formulaic interpretation of the Policy Priority Inference algorithm developed in Guerrero and Castañeda (2020, 2021).

B.1 Formulaic Interpretation

Agent and State Definitions: Consider N agents, where each agent corresponds to a policy issue i.

The state S_i of agent i is given by:

$$S_i = I_i$$

where I_i denotes the development level for policy issue i. The global state is then defined as:

$$S = (I_1, \ldots, I_N)$$

Reward and Action Function: The reward function $R_i(S)$ is expressed as:

$$R_i(S) = F_i$$

with

$$F_i = (I_i + P_i - C_i)(1 - \theta_i f_R)$$

where:

- P_i is the resource allocation to agent i.
- C_i denotes the contribution of agent i.
- θ_i indicates the event of agent i diverting funds.
- f_R is a function mapping the state of the rule of law agent to a probability.

The action A_i of agent i is defined as:

$$A_i = C_i$$

Environment Dynamics: The environment adjusts the indicator levels based on agent contributions as:

$$I_i \leftarrow I_i + \gamma(T_i - I_i)(C_i + \sum_j A_{ji} C_j)$$

Where:

- T_i is the target level for indicator i.
- A_{ji} signifies the interdependency graph.

Objective: Agents aim to devise contribution policies $C_i(t)$ in order to maximize their long-term rewards F_i. Concurrently, the central authority's responsibility is to allocate resources P_i to guide indicators towards their respective targets.

This encapsulates the primary components of the model in the cited paper using standardized terminology.

B.2 Policy Formulation and Development Strategies

Policy Priority Inference (PPI) is a powerful tool rooted in the interplay of complexity economics and computational social science. As we grapple with interconnected socio-economic landscapes and strive for strategic development planning, PPI offers precision, depth, and adaptability. Let's delve into its multifaceted utility:

Strategic Allocation & Planning: At the core of PPI is its prowess in guiding resource allocation. It allows policymakers to effectively navigate intricate policy networks, ensuring transformative resources are channeled towards areas that promise the highest impact. Furthermore, with its capability to model and reproduce stylized facts, PPI strengthens the foundation of "what-if" analyses, fostering a deeper understanding of fiscal planning and its repercussions.

Evaluative Metrics & Feasibility: PPI is not just prescriptive but also evaluative. It aids in gauging the coherence of a government's priorities relative to its overarching goals. Moreover, it provides a clear lens to assess the feasibility of a set of targets, projecting timeframes and requirements, thereby allowing for informed adjustments.

Optimization & Efficiency: The framework stands out in its ability to identify both accelerators and bottlenecks in development pathways. This dual capability facilitates the search for domains that amplify improvements across various indicators while simultaneously highlighting areas where structural constraints might impede progress. Complementing this is PPI's inherent knack for uncovering inefficiencies, ensuring that resources are utilized optimally and wastages are minimized.

Adaptability & Goal Setting: PPI's versatility is exemplified in its adaptability to diverse national contexts. Whether it's exploring a broad spectrum of developmental goals or assessing the fluidity of resource reallocation, PPI is instrumental in tailoring strategies that resonate with a nation's unique developmental narrative.

C CCDr

The CCDr algorithm introduced in this paper estimates Bayesian network structures using penalized maximum likelihood estimation and coordinate descent optimization. Here is a detailed mathematical explanation of how it works:

Let $\mathbf{X} = (X_1, ..., X_p)$ be a p-dimensional random vector that follows a multivariate Gaussian distribution with mean 0 and covariance matrix Σ. The goal is to estimate the structure of the underlying directed acyclic graph (DAG) $\mathbf{B}$ that encodes the conditional independence relationships between the variables.

We start with the structural equation model (SEM) representation of $\mathbf{X}$:

$$X_j = \sum_{i \neq j} \beta_{ij} X_i + \varepsilon_j \quad \text{for} \quad j = 1, ..., p$$

where the ε_j are independent Gaussian noise terms with variances ω_j^2. The weighted adjacency matrix $\mathbf{B} = (\beta_{ij})$ along with the diagonal matrix $\mathbf{\Omega} = \text{diag}(\omega_1^2, ..., \omega_p^2)$ define the DAG structure and noise variances.

The negative log-likelihood function based on n i.i.d. observations is:

$$L(\mathbf{B}, \mathbf{\Omega} | \mathbf{X}) = \sum_j \left[\frac{n}{2} \log(\omega_j^2) + \frac{1}{2\omega_j^2} ||x_j - \mathbf{X}\beta_j||^2 \right]$$

This function is nonconvex, so a reparameterization is done:

$$\phi_{ij} = \frac{\beta_{ij}}{\omega_j} \quad \text{and} \quad \rho_j = \frac{1}{\omega_j}$$

leading to the convex loss function:

$$L(\mathbf{\Phi}, \mathbf{R} | \mathbf{X}) = \sum_j [-n \log(\rho_j) + \frac{1}{2} ||\rho_j x_j - \mathbf{X}\phi_j||^2 \quad (5)$$

where $\mathbf{\Phi} = (\phi_{ij})$ and $\mathbf{R} = \text{diag}(\rho_1, ..., \rho_p)$. The penalized loss function is then:

$$Q(\mathbf{\Phi}, \mathbf{R}) = L(\mathbf{\Phi}, \mathbf{R} | \mathbf{X}) + \sum_{i \neq j} p_\lambda(|\phi_{ij}|)$$

where $p_\lambda(\cdot)$ is a penalty function like MCP or lasso.

The CCDr algorithm minimizes Q by performing cyclic coordinate descent. Each ϕ_{ij} is updated by minimizing $Q_1(\phi_{ij}) = \arg\min Q(\mathbf{\Phi}, \mathbf{R})$ and each ρ_j by minimizing $Q_2(\rho_j)$. After convergence, the estimates $\hat{\phi}_{ij}$ and $\hat{\rho}_j$ are transformed back to $\hat{\beta}_{ij}$ and $\hat{\omega}_j^2$. The estimated DAG $\hat{\mathbf{B}}$ is the one corresponding to $\hat{\mathbf{\Phi}}$. By using a sparsity-inducing penalty, the algorithm produces sparse DAG estimates. Theoretical results show this procedure can consistently estimate the true graph structure under certain conditions.

In summary, the CCDr algorithm is able to learn sparse Bayesian network structures by exploiting a convex reparameterization of the Gaussian likelihood and using cyclic coordinate descent with concave regularization to produce penalized maximum likelihood estimates. The sparsity helps estimate high-dimensional graphs efficiently.

D Finetuning

D.1 Social Policy Dataset

We curated a dataset derived from high-quality research papers that provide a comprehensive view of government policy across its 14 broad budgetary categories as defined in the SPOT dataset. Utilizing the SemanticScholar API, we downloaded up to 250 research papers for each category, applying filters for language and citation count. Our final dataset, after removing duplicates, comprises 1450 research papers. During preprocessing, the text was segmented into spans ranging from 128 to 256 characters, with a 35% overlap. Only English-language papers were retained. Any textual inconsistencies arising from PDF to text conversion were rectified using 'stabilityai/StableBeluga-7B'. The dataset is open-sourced and available at this repository.

D.2 Instruction Tuning Dataset

The inherent methodology of our CPUQ approach necessitates a response style typical of instruction-tuned language models. This specific response mechanism aids in understanding and generating appropriate answers for Prompt + Answer scenarios. The Social Policy Dataset contains continuous prose, from which a language model learns continuation, as opposed to instruction following. To ensure our model retains a strong ability to respond, we integrated the WizardLM dataset (Luo et al., 2023b; Xu et al., 2023; Luo et al., 2023a). This dataset bridges the instructional response gap, fortifying our model's ability to handle the nuances of our CPUQ prompting approach.

D.3 Finetuning Setup

Our finetuning setup employed QLORA with double quantization and an Adam optimizer (lr=1e-3, b1=0.9, b2=0.95). We applied a constant schedule with a 200-step warm-up and distributed over 6 RTX3090s. For the 7bn models, we used a batch size of 30, while for the 13bn models, the batch size was 18, with gradients accumulated over 3 steps, resulting in an effective batch size of 54. An innovative paired early stopping rule was designed, halting the process if no improvements are detected on validation sets for either instruction or next token prediction tasks.

E CPUQ: Further Considerations

Constraints: One important constraint of this framework is that, when using the categorisation methodology, the user must specify that the categorical ouputs chosen be numbers and not letters. An intuitive explanation for this is based on the idea of ensuring that the probability of the next token is only focused on the probability of selecting a correct categorical number and not also predicting a general continuation. For example, suppose we ask a LM to answer the Question: "Choose the category letter that best answers the question: Which is the most environmentally friendly form of transport for people in a large city: A) SUV, B) Bus or C) Bike. The ideal set of responses would be ["A.", "B.", "C."]. However, due to the unconstrained nature of Language Models the set of responses also includes sentences such as ["*A* likely answer to this question would be C", "*B*ased on Bikes having no emissions "C" would be the correct category."]. Initial experiments indicated that the extent to which this is a problem is more tied to the language model strength than the phrasing used in the prompt.

Excluding an NA from Categorical Answer Space: In our work, we use a binary categorization for our 'Yes', 'No' prediction and exclude a third option which could reflect a non-committal or uncertain prediction. Specifically, the two alternatives for this category are 'I don't know' and 'I am not sure'. The difference between these phrases can have implications both in interpretation and in practical implementation. If we were to extend the categorical answer space to include a third category, our set of answers would look like ['Yes', 'No', 'I don't know' / 'I am not sure'].

We begin by discussing the category 'I am not sure'. The category 'I am not sure' implies a more comprehensive form of uncertainty compared to 'I don't know'. Not only does it suggest a lack of knowledge, but it can also technically include a distribution over 'Yes' and 'No'. For instance, stating 'I am not sure' might imply that one is 20% certain of 'Yes' and 80% certain of 'No'. This makes the categories not strictly mutually exclusive. However, this comprehensive interpretation presents its own problems. When a probability is assigned to a category like 'I am not sure', we are essentially quantifying uncertainty about uncertainty.

Now, considering the simpler 'I don't know' option, from a theoretical standpoint, it represents an acknowledgment of one's epistemic boundaries on a topic, without necessarily implying any specific probability distribution over 'Yes' and 'No'. This does not pose a logical problem. However, in practice, we encountered an issue: for cases where the correct answer to a categorical question was 'No', language models were inclined to allocate a high probability to 'I don't Know'. This tendency meant that 'No' and 'I don't know' cannibalized each other's assigned probability, complicating the mapping of probabilities to categories.

The nuanced difference between the two categories and the inherent difficulties they bring to the table resonate with the Knightian distinction between risk and uncertainty, where some events inherently defy easy probabilistic characterization (Knight, 1921). Arrow's critique on the limits of decision-making under uncertainty complements this, indicating potential shortcomings of standard decision models in scenarios with intertwined uncertainty levels (Arrow, 1971).

To conclude, while 'I don't know' is a straightforward acknowledgment of lack of knowledge, adding a probabilistic layer to it leads to contradictions, especially when the boundaries between the categories blur.

F Economic Allocation Dataset

The dataset can be composed into three parts

1. Dataset mapping broad government budget items and indicators, annotated by experts
2. Timeseries of United Kingdom's spending across 32 fine-grained government budget items
3. Timeseries of 258 population health and socio-economic indicator levels in the UK

1. Government Spending Timeseries: We create a dataset showing local authority expenditure over 32 fine-grained UK budget items. After preprocessing, we keep data between 2013 and 2019. To retrieve this data, we draw upon the Spend and Outcomes Tool (SPOT) (Office for Health Improvement & Disparities, 2023), created by the Office for Health Improvement and Disparities (OHID, Department of Health and Social Care, England). In terms of expenditure, SPOT includes net current local authority revenue expenditure and financing. We focus on this fraction of the total Public Health Funding as local authorities have a relative leeway to allocate resources to fund Public Health Services, as opposed to the expenditure earmarked to cover National Health Service (NHS), primary care, prescribing, and other staff costs. It is also smaller than other types of expenditure available to local authorities, such as Education, which is much larger but more rigid in the services to allocate.

2. Population Health and Socio-economic Indicator Timeseries: In terms of health service provision and population level health outcomes, we obtain data from Fingertips (for Health Improvement & Disparities , OHID), which is a large dashboard of health-related information reported by different public entities and organised into themed health profiles. The Consumer Price Inflation timeseries (for National Statistics , ONS) and the mid-year estimates of resident population are obtained from the UK Office for National Statistics. Rule of law and Quality of Monitoring parameters to be employed in the PPI algorithm were obtained from the World Development Indicators.

3. Related Broad Budget Item and Indicators Dataset In total there are 258 unique indicators and 15 broad budget items. SPOT provides a dataset indicating which broad government budget items are intended to effect which indicators.

G Scaled Entropy For Categorical Distribution

In this section, we discuss the scaled entropy for Categorical Distributions, emphasizing its similarities with the traditional normalization method.

The key properties of the scaled entropy for Categorical Distributions are:

1. The entropy is scaled to the range [0, 1], making it comparable across distributions with different numbers of categories.
2. The surprisal is consistent across different distributions.
3. For a uniform distribution over n categories, the scaled entropy is always 1, providing an intuitive measure of maximum uncertainty.
4. The method is specifically tailored to categorical distributions, offering a direct and intuitive comparison between distributions.

To draw parallels between the two normalization methods, consider the entropy formula with base n:

$$H(X) = -\sum_{i=1}^{n} \frac{1}{n} \log_n \frac{1}{n}$$

Given that $\log_n n = 1$, the entropy for a uniform distribution simplifies to:

$$H(X) = 1$$

This is analogous to the traditional method of dividing by $\log_2(n)$, where the entropy of a uniform distribution is also normalized to 1. The primary similarity is that both methods aim to scale the entropy value to a range of [0, 1], ensuring comparability across different distributions.

Benefits of using the number of categories n as the base for normalization include:

- Direct and intuitive comparison between distributions with different numbers of categories.

- The entropy value provides a clear indication of the distribution's nature, with 1 indicating a uniform distribution and values close to 0 indicating deterministic distributions.

Another advantage of this normalization method is its simplicity and ease of interpretation, especially for audiences not deeply familiar with traditional information theory concepts. This is crucial since our focus is on Economic Allocation systems, which could include policymakers. In this context, this measure of uncertainty offers an easily interpretable value between 0 and 1.

H Reproducibility Statement

Code: The code and data used in this study can be found at this repository [Redacted for Review].