CPUQ: Categorical Perplexity Based Uncertainty Quantification with Language Models

Anonymous ACL submission

Abstract

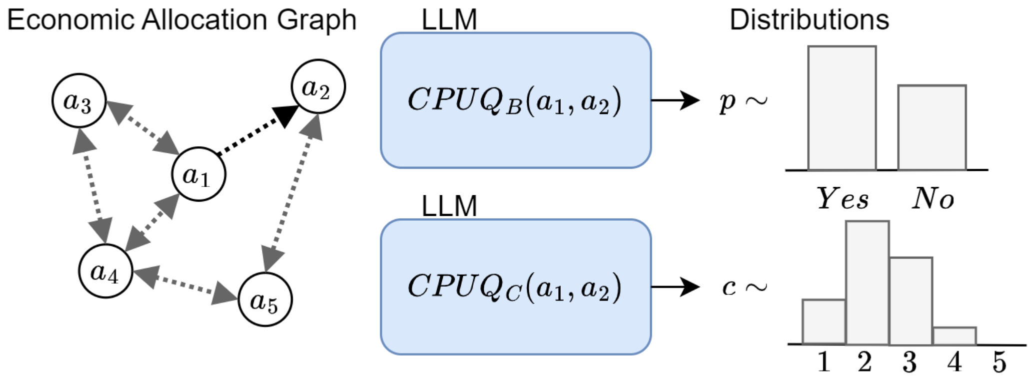
Agent Based Modelling (ABM) algorithms for Economic Allocation (EA) systems model interactions between economic agents and indicators. These EA-ABMs provide important insight for policy makers and decision analysis as they can be used to model complex systems such as Government Spending or Financial Market Contagion. However, the utility of EA-ABM's depends on the quality and interpretability of the underlying graph's estimated edge weights. Statistical network estimation methods perform poorly due to these datasets often having limited timesteps of data but a large number of nodes (economic actors or indicators) and edges (causal relationships). We propose a structured method to use Large Language Models (LLM) to produce predictive hurdle distributions for the edge weights; enhancing interpretation through uncertainty quantification and textual reasoning. Our approach, Categorical Uncertainty based Uncertainty Quantification (CPUQ) decouples the modelling of causal relationships into separately modelling existence and causal relationship strength. Through evaluation on a real Economic Allocation dataset, we show that CPUQ produces probabilistic predictions well aligned with experts opinions, and achieves better EA-ABMs forecasting ability than existing statistical and LLM based methods. We also motivate a solution for the issues of conflating a language model's uncertainty with syntactical uncertainty as opposed to semantic uncertainty.

1 Introduction

Economic Allocation (EA) Agent Based Modelling (ABM), crucial for simulating economic allocation processes, can be hampered by the complex challenge of determining the edge weights for each node pairing within vast graphs representing economic actors and economic/financial indicators. In many situations, these graphs can encompass edges numbering in the order of 10^6, each demanding precise indications of relative strength or uncertainty. This becomes even more complex since the number of nodes (agents) and potential edges (interactions) often dwarfs the span of data available in graphs underpinning Economic Allocation Systems. There exist several statistical methods for estimating directed networks, each with different assumptions and limitations. For example, Bayesian networks methods (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 as indicated in (Castagneto-Gissey et al., 2014) and are inappropriate for test of predictability involving more than two variables. Further, the these methods often require sufficient observations-to-variables ratio, a common limitation in many Economic Allocation Systems even with matrix factorization methods. (Aragam and Zhou, 2015) propose a non-convex optimization approach that extends Bayesian network methods to graphs where p » n, overcoming the necessity for sufficient observations-to-variables ratio.

Previous works (Yamasaki et al., 2023; Bansal et al., 2019; Saxena et al., 2022) have highlighted the effectiveness of network estimation using Language Models (LM) on Textual Attribute Graphs, where each node is represented by some descriptive text. Furthermore, these Language Models can also be extended to producing well calibrated probabilistic predictive distributions for the relationships between two entities as recent works in Language Model Question Answering (Kuhn et al., 2023; Kadavath et al., 2022) have shown.

However, many of these LM based approaches to calibrated distributions focus on the simpler tasks of distributions over categorical outputs (Kuhn et al., 2023; Jiang et al., 2021), as opposed to expressive probabilistic predictions over ordinal and



$$\text{weight}(a_1, a_2) \sim \mathbb{1}[p{=}yes > h] \times c$$

$$\text{conditional uncertainty}(a_1, a_2) = 1 - \text{scaled entropy}(c)$$

Figure 1: CPUQ: This diagram shows the CPUQ methodology for determining predictive distributions for edges in a Textual Attribute Graph, modelling interactions between economic agents a_i. $CPUQ_{B,C}$ is an LLM agnostic method that can produce a Bernoulli or Categorical Distribution determining edge existence and conditional edge weight respectively. The weight of the directed edge between a_1 and a_2 is then a hurdle mixture distribution. The conditional uncertainty for an existing edge is then based on entropy where the log base is equal to 5, the number of scale categories.

numerical outputs. With the latter requiring more complex properties of for an output distribution to be consistent such multi-modality or convexity.

To tackle these challenges our approach, Categorical Perplexity based Uncertainty Quantification (CPUQ), is designed to estimate edge weights in Text Attribute Graphs (TAG), graphs where each node can be represented by textual information. Our approach outputs a zero-inflated mixture distribution, which includes a Bernouilli distribution to model the chance of no edge existing between two nodes and a Categorical distribution to model the weight of the edge if it does exist. Relative to statistical network estimation methods the use of text attributes better reflects causation modelling. Furthermore, our method also provides an interpretable textual explanation for the output provided and the use of predictive hurdle mixture distributions promotes sparse networks while limiting inference time.

In this work we validate and evaluate our method on a Economic Allocation system where a regional government (United Kingdom) must allocate its budget between many budget items with a goal of achieving specific levels for a set of indicators over a 9 year horizon.

In developing CPUQ, we also analyse conceptual and practical issues with uncertainty quantification with language models. We further investigate any biases induced by our uncertainty quantification approach by inspecting the distribution of edges predicted relative to existing approaches. key benefits of our method encompass its strong alignment to human labelled datasets, interpretability, cost-effectiveness, automation potential and ability to perform uncertainty quantification. For instance, explanations for specific edges are integral to our inference process.

The essence of our contributions lies in:

- Develop CPUQ, which outputs a interpetable hurdle categorical distribution through categorical style questions.
- Provide a formal motivation for CPUQ, highlighting the complexity of semantics and syntax when performing perplexity based Question Answering.
- Show CPUQ methods produce strong alignment to expert annotations for the causal edges in a graph underlying an real world Economic Allocation System.
- Demonstrate that CPUQ based network estimation performs outperforms existing statistical and Language Model network estimation methods when evaluated by the performance of an Economic Allocation Agent Based Modelling algorithm.

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 is the method of prompting a language model with phrases/synonyms which have the same meaning and observing the variation in output. A large body of recent works (Arora et al., 2022; Wei et al., 2022) have demonstrated strong performance increases on QA tasks through designing methods to find an optimal prompt. This line of research would suggest an optimal prompt exists, and prompt variation does not test a model's predictive uncertainty but mostly the quality of the prompt. Further, (Jiang et al., 2021) showed the prompt specification becomes less important as the foundational models become better calibrated.

Sequence perplexity based measures: The probability of the a text sequence, s, is the product of the conditional probabilities of new tokens given past tokens, whose resulting log-probability is $\log p(\mathbf{s} \mid x) = \sum_i \log p\left(\mathbf{s}_i \mid \mathbf{s}_{<i}\right)$, where $\mathbf{s}_i$ is the i'th output token and $\mathbf{s}_{<i}$ denotes the set of previous tokens. From this distribution, previous works (Jiang et al., 2021; Kuhn et al., 2023) have used the corresponding predictive entropy $H(s \mid x) = -\int p(s \mid x) \ln p(s \mid x) dy$ as a point statistic of uncertainty. Alternatively, (Malinin and Gales, 2018; Murray and Chiang, 2018) used the arithmetic mean log-probability $\frac{1}{N} \sum_i^N \log p\left(s_i \mid \mathbf{s}_{<i}\right)$.

Previous works (Kuhn et al., 2023) have briefly stated the lack of "theoretical justification" for this method. We expand upon this argument and propose a formal condition that holds true when the output space includes tokenized sequences s with length over 1.

When considering sequences over length 1, the conditional probability $p\left(\mathbf{s}_i \mid \mathbf{concat}(x, \mathbf{s}_{<i})\right)$ 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.

We believe previous works have highlighted specific incidences of this condition. For example, (Murray and Chiang, 2018) highlights 'label bias'; a models' stylistic bias towards a specific length of response which reduces the relative likelihood of longer answers. While other works, (Jiang et al., 2021; Kuhn et al., 2023) show a language models bias towards different styles of expressions with the same 'semantic equivalence' must be taken into consideration.

Overcoming Stylistic Bias When the response space for a language model is constrained to a set of token sequences of maximum length 1, the scope for syntactic style to influence the output distribution is limited. Intuitively, this is reflected by the singular 'style' of response when answering a Yes/No question e.g a respondent replies 'Yes.' or 'No.' independent of any syntactic style they may have. We can express this by decomposing the conditional probability of an output sequence s, given prompt x, $p(\mathbf{s} \mid x) = \sum_i \log p\left(s_i \mid \mathbf{concat}(x, \mathbf{s}_{<i})\right)$ into a joint conditional probability involving a latent semantic meaning $m \in M$.

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

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

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

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

Where equation 2 decomposes the surface realization, s, distribution into a 2 step process of initially modelling the semantic meaning m, then conditionally modelling s over a bi-variate distribution over prompt x and the output's latent semantic meaning m . The simplification in Equation 3 is due to the constraint to 1 token responses. Finally, in equation 4, we constrain our prompt x to a prompt set $x \in X'$ for which the output distribution will be heavily weighted on a unique token s^*, $p(\mathbf{s}_0 = s^m \mid x, m)$ for each possible semantic meaning m.

As the $p(\mathbf{s}_0 = s^* \mid x, m)$ approaches 1 for $m \in M$ the primary source of variability in the $p(\mathbf{s} \mid x)$ can be attributed from the term $\log p(m \mid x)$. This emphasizes that, in the restricted case of single-token responses to a specific set of prompts $x' \in X$, the model's uncertainty predominantly originates from the latent semantic meanings rather than the stylistic variations of the response.

We propose to satisfy these conditions with **Categorical Question Style** prompts.

3 Categorical Perplexity based Uncertainty Quantification

In Section 2 we motivated the use of Categorical Prompts for Uncertainty Quantification (CPUQ) as more efficient than sequence sampling methods while correctly providing uncertainty over distinct semantic outputs instead of syntactic outputs.

We remind the reader that our downstream task is network estimation in Textual Attribute Graphs underpinning 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 three steps:

1. Determining Edge Existence
2. Determining Edge Weight
3. Determining Predictive Uncertainty

1. Determining Edge Existence CPUQ$_B$ For edge existence we create a categorical question style prompt that requires the model's response to be the number token for the correct category number. We then use perplexity over the one token output space to create a Bernoulli distribution over the corresponding 'Yes' edge exists or 'No' edge doesn't exists answers.

2. Determining Edge Weight CPUQ$_C$ Given the edge existence probability reaches a specific threshold hurdle h, we create a categorical question style prompt that requires the model's response to be the single token for the number representing the relationship strength between two economic agents / indicators on a scale. This interval must only include single digits and in this work we choose integers between one and five. After attaining the categorical distributed c, illustrated in Figure 1, we determine the categorical mean for the weight by multiplying each value by a normalized likelihood as shown in Equation 6.

$$p_{norm}(s^i \mid x) = \frac{f(s^i \mid x)}{\sum_j f(s^j \mid x)} \tag{5}$$

$$\mu(s) = \sum_{i=1}^{10} s^i \cdot p_{norm}(s^i \mid x) \tag{6}$$

A benefit of the hurdle h, is that it reduces the computational expense required by having to perform this secondary weight determination step on all samples.

3. Determining Predictive Uncertainty Point statistics for uncertainty over edge existence or uncertainty over edge weight can be determined using an entropy based measures. Focusing on maximizing interpretability of this method for policy makers / decision makers, we move away from previous works (Kuhn et al., 2023; Jiang et al., 2021) which simply used entropy, and instead use a normalized entropy measure for categorical distributions which uses a base equal to the number of categories and inverts the value such that 1 implies maximal certainty and zero implies maximal uncertainty. In Appendix G we provide a brief motivation for the use of our proposed normalized entropy measure. For the Bernoulli distribution of edge existence, the normalised entropy measure $H(B(p))$ is given by:

$$H(B(p)) = 1/2 \cdot (-p \log_2 p - (1-p) \log_2 (1-p)) \tag{7}$$

For the Categorical distribution pertaining to edge weight, assuming the edge exists, the normalised entropy $H(M)$ is defined as:

$$H(M) = -\sum_{i=1}^{5} \frac{1}{5} p_i \log_5 p_i \tag{8}$$

Here, p_i is the probability of the edge weight being the i-th value.

Unbiasing Categorical Label Order In initial experiments we observed indications of stylistic bias existed towards either the first or second categorical response, e.g. consistently inflating the probability assigned to category answer 1) Yes. To prevent this we introduce a method which asks the same question twice with the order of the categorical responses switched, following this we average the two distributions.

Question w/ Reasoning Previous works (Wei et al., 2022; Zhang et al., 2022; Wang et al., 2023) have demonstrated improvements to language model predictive ability when the language model is prompted to break its deductive process into intermediary steps. We experiment with a version of CPUQ that prompts the language model to produce an intermediary explanations, prior to producing its categorical answer. This is the final method presented in Figure 1.

Fine-tuning We follow previous works (Jiang et al., 2021) exhibiting the benefit of fine-tuning on domain specific knowledge. In our experiments

Prompt Stage	Prompt Template
Question w/ Reasoning	PROMPT 1: 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}? LLM: [Model's response]
CPUQ_B Output	PROMPT 2: Write only the number of the category that fits the following statement. "Statement: [Model's response]" 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 Output	PROMPT 3: 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 causal relationships in our Economic Allocation experiments. The {indicator} placeholders represent textual representations of indicators. {effect_type} can be 'direct', 'indirect', or blank. $\mathrm{CPUQ}_{B/C}$ 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 methods.

we fine-tuned models under 17bn parameters in size. Due to hardware limitations, larger models were not considered. To fine-tune these models, we used both an instruction dataset and a curated knowledge-focused free-flowing text dataset on Social Policy. The model was trained on both datasets in equal proportions. This approach aims to enrich the model's expertise in Social Policy while retaining its inherent instruction-following capabilities which ensure that the conditional distribution $p(s \mid x)$ has the majority of its mass on tokens correlating to a response categorical answer as opposed to tokens which would be continuing the text. We provide more information on these datasets in Appendices D.1D.2.

4 Validation: Alignment To Expert Annotation

In this set of experiments, we validate the degree of calibration of our approach by investigating its ability to align to 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 fully detail the dataset in Appendix F. The part of the dataset used in this validation experiment provides pairs of (broad budget item, indicator) for which the broad budget item does affect the indicator. In total, after pre-processing, there are 258 unique health indicators allocated to one of 15 broad budget items. We use negative sampling to produce negative samples for this dataset e.g. pairs of (budget item, indicator) for which the budget item does not affect the indicator.

Model. We use language models from the llama family (Roumeliotis et al., 2023b,a). We experiment model with sizes of 7bn, 13bn and 30bn parameters.

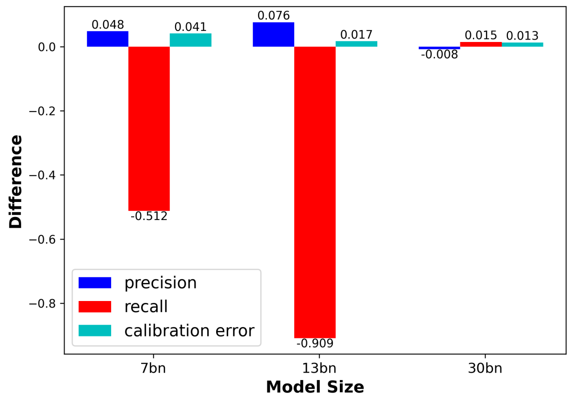
Baselines For baseline methods we include two approaches (verb_open) and (verb_closed) which utilise a similar method to (Tian et al., 2023; Zhou et al., 2023; Lin et al., 2022), which simply prompt the model to verbalize its answer with an open-ended or close-ended response. For a baseline model we compare our method against gpt3.5-turbo, a strong performant model. This provides an interesting insight into the effect of foundational model strength.

Results As this is a binary classification task we present F1, Precision and Recall scores in Table 3. We notice that our method performs competitively with the verbalization approaches which do not produce probabilistic outputs. The CPUQ Question w/ Reasoning outperforms the CPUQ Closed Ended Question, highlighting the benefit of encouraging the model to utilize its own reasoning. GPT3.5 provides the strongest performance highlight the significance of foundational model strength.

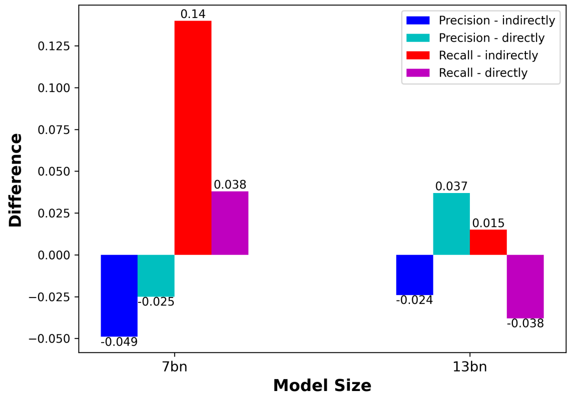
Ablation Experiments In these experiments we also include the Expected Calibration Error (ECE) metric, introduced by (Guo et al., 2017), quantifies the calibration quality of probabilistic predictions. It computes a weighted average of the differences between observed accuracy and the predicted confidences across distinct buckets or intervals.

To address stylistic bias in categorical label order for the CPUQ$_B$ method we found that recall experiences a significant degradation for foundational models of size 13bn and below, whereas the 30bn parameter model experiences modest performance increases across recall and Expected Calibration Error. This implies that the smaller foundational model's slightly struggle when asked categorical Yes/No questions where the arrangement of answers is in an unconventional order such as 1) Negative Response 2) Affirmative Response.

For both the 7bn and 13bn model sizes we observe a decrease in precision when 'indirectly' is introduced to the prompt, reflecting the notion that the language model may be factoring in loose relationships when compared to the expert annotators judgement. On the other hand, the Recall increases across both sizes when 'indirectly' is introduced to the prompt, reflecting the complementary notion the language model's more loose interpretation of what constitutes a relationship allows less chance of missing possible relationships.



(a) Unbiasing Categorical Label Order



(b) Varied Effect Type

Figure 2: **Ablation Experiments:** These figures represent predictive performance when classifying the edge existence in Textual Attribute Graph underlying an Economic Allocation dataset involving U.K. government spending and socio-economic indicators. Figure a) presents the changes in predictive scores when we implement our method to unbias categorical Label order, explained in Section 3. Figure b) presents the performance change from specifying the prompt templates' "effect type" as 'directly' or 'indirectly' when compared to having no specification of relationship type between spending on a government budget item and a socio-economic indicator. The prompt template is exemplified in Table 1.

5 Evaluation: EA-ABM Forecasting

We compare the forecasting performance of an EA-ABM algorithm called Policy Priority Inference (PPI) when the underlying graphs is estimated using our CPUQ methods and other baseline methods. For each method/graph, 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.

For a detailed explanation of the PPI algorithm please refer to Appendix B. The PPI algorithm models two levels of interactions. The first is the budget item to indicator (b2i) interaction set, representing the 1st order effects of government spending and the indicator to indicator (i2i) interactions capturing the second order spillover effects. In the PPI algorithm the b2i edges are binary, while the i2i edges are floats, appropriate for our CPUQ$_B$ and CPUQ$_C$ methodologies respectively.

Data. We have 7 years of annual data for government spending on the fine grained health related budget items and the levels of socioeconomic-health indicators. The first 5 years form the training set. The final ttwo years form the test set. There are 32 fine-grained budget items and 258 socioeconomic-health indicators. This means there are 8256 possible b2i edges and 66564 possible i2i edges for estimation. Appendix F provides more detail explanation of the dataset used.

Baseline Methods. Each experimental result consists of methods for predicting both b2i and i2i edges independently. For determining the b2i edges, baseline methods include verbalization with close-ended questions as detailed in Table 1 and naive expert annotation (ea). 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$(b_b, i){\rightarrow}(b_f, i)$.

For determining i2i edges, baseline methods encompass zero (representing no spillover effects between indicators), verbalization 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 Section 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 and D.2.

5.1 Results

For the set of experiments where the i2i methodology is fixed to naive expert annotation (n.a.e.) and b2i method varies, in Table 2 we observe that the CPUQ$_C$ performs competitively with verbalization and that the CPUQ$_C$/verbalize method achieves the highest mse/mae score.

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. However, 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	<u>0.04814</u>
n.e.a	CPUQ$_C$	<u>0.01195</u>	0.04820
verbalize	verbalize	0.01211	0.04830
CPUQ$_B$	CPUQ$_C$	<u>0.01202</u>	<u>0.04825</u>

Table 2: **PPI Forecasting Performance:** Prompting methodologies are varied for prediction of binary budget item to indicator (b2i) and non-binary indicator to indicator (i2i) causal relationships. For b2i edges, methods include naive expert annotation (n.e.a) and verbalization. Float i2i methods include zero (no spillover), verbalization, entropy from CPUQ$_B$ with > 0.5 probability, and the CCDr algorithm. Results highlight the competitive performance of CPUQ$_C$, but also the increased relative difficulty LLM models have labelling binary valued edges.

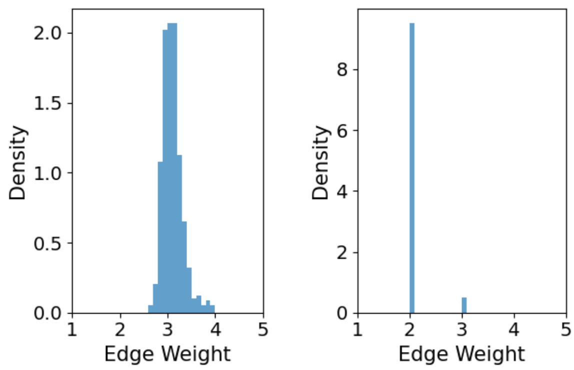
The second set of experiments focus on also predicting the binary b2i edges in the graph as well as the non-binary i2i edges in the graph. We notice that our CPUQ outperforms the verbalization method.

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 graph. The verbalization method suffers from the output being limited to producing on 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.

6 Related Work

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 do not directly rely on the standard probabilistic measure of perplexity.



(a) CPUQ_C (b) Verbalization

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 verbalization strategy when using the same underlying language model. We notice the verbalization 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.

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 finetune predictors on the embeddings of generating models to predict models uncertainty. While promising, these approaches need task-specific labels, additional training, and seem to be unreliable out-of-distribution (Kadavath et al., 2022).

Some prior work has addressed the important concern of grouping semantic similar terms when distributed probabilities over candidate answers. (Jiang et al., 2021) address 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 at they require a secondary model which is used to evaluate similarity of different candidate answers and also utilise 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.

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 3: **Expert Annotation Alignment:** Evaluation of predicting the influence of local government budget items on socio-economic indicators using different prompting methodologies. Compared are the CPUQ methods against GPT3.5 and verbalization strategies. Verb_closed and verb_open elicit deterministic Yes/No answers, while CPUQ methods produce probabilistic outputs. Examples of Prompt Styles are in Table 1. The 30bn model is a derivative of the llama language model family. Q.R. denotes Question w/ Reasoning. $CPUQ_C$ performs competively with verbalization, while achieving significantly stronger recall.

7 Conclusion

We introduced CPUQ, a novel method for uncertainty quantification using Language Models. This method utilizes categorical-style questions to generate insightful hurdle categorical distributions for edges in a textual attribute graph associated with Agent-Based Modelling for Economic Allocation. Validated against a U.K. dataset on government spending and socio-economic indicators, CPUQ not only aligns effectively with expert annotations but also outperforms prominent alternative LLM and statistical methods. Critically, it can deliver accurate and interpretable distributions over edge weights estimations vital for network estimation in Economic Allocation systems used by policy makers and decision makers.

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 policy makers and decision makers. Therefore, we advocate for morally correct and responsible practices in the case of real-world application.

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.

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.

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.

Konstantinos I Roumeliotis, Nikolaos D Tselikas, and Dimitrios K Nasiopoulos. 2023a. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Konstantinos I Roumeliotis, Nikolaos D Tselikas, and Dimitrios K Nasiopoulos. 2023b. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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 intricate 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 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 policies 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 brief 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 $R_i(S)$ for agent i 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 Developmental 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 advancements, 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 observable fiscal patterns, 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 set 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 resource 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:

$$L(\mathbf{\Phi}, \mathbf{R}|\mathbf{X}) = \sum_j [-n \log(\rho_j) + \frac{1}{2}||\rho_j x_j - \mathbf{X}\phi_j||^2 \qquad (9)$$

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. 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 towards learns continuation, as opposed to responding. To ensure our model retains strong 'response style', 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 PUQ prompting approach.

D.3 Fine-tuning Setup

Our finetuning setup employed QLORA with double quantization, 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: Important constraints of this methodology are that when using the categorisation methodology the user must specify that the categorical numbers 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 LLM 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 C", "Based on Bikes having no emissions "C" would be the correct category.]. Initial experiments indicated experiments 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 opt out of 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 indicating related broad government budget items and indicators, annotated by experts
2. Timeseries of United Kingdom's Spending across 32 finegrained Government Budget Items
3. Timeseries of 258 socio-economic indicator levels in the U.K

1. Government spending timeseries We create a dataset showing Local Authority expenditure over 32 finegrained UK budget items. After post-processing 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, often referred as Revenue Outturn 3. 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. Socioeconomic 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 time series(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 governance 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 which indicates which broad government budget items are intended to effect which indicators.

G Normalised Entropy For Categorical Distribution

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

The key properties of the normalised 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 normalized 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 policy makers. 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].