Generative Brand Choice*

Kevin Lee
University of Chicago, Booth School of Business

May 7, 2025

Abstract

Predicting demand for new products is an important and challenging problem in marketing, especially for product categories where brand is a key driver of choice. In these settings, observed product attributes do not explain choice patterns well, which makes predicting sales of a new product as a function of marketing mix variables intractable. To address this problem, I develop a scalable framework that enriches structural demand models with large language models (LLMs) to predict consumer preferences for new brands. After estimating preferences for existing brands using a structural model, I use an LLM to learn a mapping from text descriptions of the brand and consumer to these estimated preferences. When fine-tuned in this way, I show that the tuned LLM is able to *generalize* to previously unseen brands that were excluded from the training sample. In contrast, conventional models based on text embeddings return predictions with zero or even negative correlation with the actual utilities. My fine-tuned LLM achieves the first informative predictions of consumer preferences of new brands from raw text, with a 25% smaller mean squared error and correlation between predictions and held out preferences of 0.34. Additionally, I combine causal estimates of the price effect from instrumental variables methods with the LLM predictions to enable pricing-related counterfactuals. More broadly, this approach illustrates how new kinds of questions can be answered by using the capabilities of modern LLMs to systematically combine the richness of qualitative data with the precision of quantitative data.

*Email: kevin.lee@chicagobooth.edu. I thank Sanjog Misra, Eric Budish, Pradeep Chintagunta, and Giovanni Compiani for helpful discussions and support. This paper benefited from discussion with Malika Korganbekova and Walter Zhang. Researcher's own analyses calculated (or derived) based in part on data from Market Track, LLC dba Numerator and marketing databases provided through the Numerator Datasets at the Kilts Center for Marketing Data Center at The University of Chicago Booth School of Business. The conclusions drawn from the Numerator data are those of the researcher and do not reflect the views of Numerator. Numerator is not responsible for and had no role in analyzing or preparing the results reported herein. The raw Numerator data was used solely to estimate consumer preferences using a standard economic choice model. At no point did researchers use any AI tools with Numerator's raw data.

1 Introduction

I develop a general framework for modeling brand preferences from text data by combining a structural demand model with a large language model. My motivation is the well-known and unsolved problem of predicting demand for new brands, which is especially difficult in product categories where brand is a major driver of choice (e.g. cars, electronics, packaged goods). In these settings, observable product attributes do not explain choice patterns well. Brand fixed effects can capture the effects of unobservable attributes for existing brands, but they require historical data to estimate and are not known for new brands. As a result, demand models do not give informative predictions for new brands or changes to existing ones.

I propose a solution to this problem that augments a structural demand model with

(i) text data,

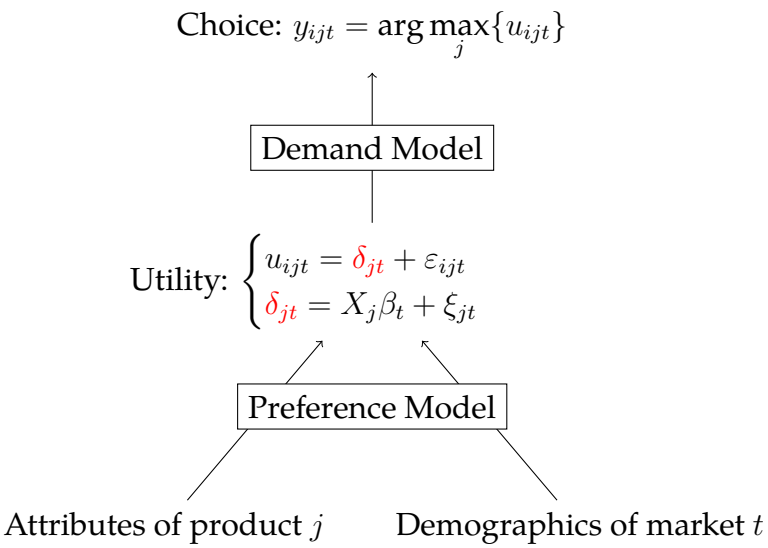
(ii) a large language model (LLM),

(iii) and a novel method for adapting the LLM to the structural model.

Importantly, (i) or (ii) on their own fail.

My method is applicable to any demand estimation setting where unobserved product attributes are relevant for consumer preferences and no historical data is available to estimate the effects of unobservables. It provides relevant guidance for decisions on positioning a new brand, brand extension (e.g. Chobani-branded oat milk), or brand crossover (e.g. Febreeze-scented Tide detergent). I will also show how to predict what kinds of customers will like a new brand and solve for the profit-maximizing price.

Traditional demand models would answer these questions by modeling demand as a function of product attributes, price, and consumer demographics that proxy for preference heterogeneity (Lancaster (1966), McFadden (1974), Berry (1994), Berry et al. (1995)). For choice data aggregated at the market level, these models look like:



Consumer i in market t values product j at utility u_{ijt} and chooses the product j with the highest utility. It is common to denote the "systematic" part of the utility by δ_{jt}. This is the average utility for product j by consumers in market t and the main object of study in this paper. It can be estimated from market shares and a structural model (Berry (1994), Berry et al. (1995)). The estimated δ_{jt}'s are decomposed as a function of product characteristics X_j, with preference heterogeneity modeled according to demographic variables. The ξ_{jt} term denotes the component coming from unobserved product attributes. For many product categories, the observed product attributes X_j (e.g. size/MPG/horsepower for cars, caffeine/calories for energy drinks) and demographics (e.g. age, gender, household size) are inadequate for predicting δ_{jt} without the addition of a brand fixed effect to capture the effect of unobservables. This creates an issue when estimating demand for products from new brands as the fixed effect is unknown.

Previous work has attempted to capture the effects of unobservables by using fixed effects of existing brands or by gathering additional data. Taking the fixed effects of existing products, as in Berry et al. (2004), is reasonable in some scenarios like a brand extension but not generally applicable. Commonly collected forms of additional data are consumers' ratings of brands on perceptual dimensions that are deemed important for modeling their preferences, e.g. the "luxuriousness" of a car brand, or the "mushyness" of cereal (which

was shown to be helpful in Nevo (2001)). More examples are discussed in Kamakura and Russell (1993) and Walker and Ben-Akiva (2002). This approach is effective to the extent that an analyst can choose the right dimensions to measure, and it also requires costly data collection. Consumers must rate all dimensions for all existing brands as well as all hypothetical new brands. A subtle but arguably more important issue is that reducing brands to dimensions that an analyst specifies in advance will be overly restrictive and discard information that would have been useful for predicting δ_{jt}. This restrictiveness means that even when data is abundant, there is an upper bound on the best possible performance.

To avoid the need to choose the correct perceptual attributes to measure, I propose to model brand preferences as a function of text descriptions of brands and consumers. In principle, text descriptions of brands should contain information that is useful for modeling preferences, and text descriptions of consumers should be informative about preference heterogeneity. I show how to incorporate this textual information directly using an LLM, thereby avoiding the loss of information from intermediate coding steps.

My method consists of four steps:

1. Estimate δ_{jt} for existing brands and markets with standard methods.

2. Collect text descriptions of existing brands and markets.

3. Fine-tune LLM to predict the δ_{jt}'s from Step 1 given text from Step 2.

4. Use the tuned LLM to predict δ_{jt} out of sample on new brands or markets.

The important part is Step 3. I then compare my method against several natural benchmarks: demand models based on observable attributes (i.e. not using text), using a general-purpose LLM without any fine-tuning, and existing machine learning methods for quantifying text based on text embeddings. On existing brands, I find that text data improves predictions of δ_{jt} compared to not using text data, achieving a 33% reduction in mean squared error. Both standard machine learning methods using text embeddings and my

method using a tuned LLM perform well. On previously unseen brands, text data is again helpful, but my method yields much more informative predictions than machine learning models based on embeddings. When comparing actual and predicted values of δ_{jt}, models based on embeddings return predictions that have zero or negative correlation with the actual values. This is because the observed data lacks the variation required to learn how consumers value newly varying dimensions of the embedding. In contrast, leveraging a pretrained language model as a prior and then fine-tuning yields predictions with a correlation of 0.34. The Spearman correlation shows the same pattern, which indicates the usefulness of the fine-tuned model in ranking which markets will like the previously unseen brands the most. Accuracy in ranking which markets prefer the new brand the most offers a systematic way to select markets to expand into (see e.g. Bronnenberg and Mela (2004)). I also show how to leverage the structural model to solve for the optimal price. The pricing calculation combines predictions from the tuned LLM with causal estimates of price effects from instrumental variable methods, further illustrating how existing economic methods can be enhanced by technical innovations in AI.

Next, I investigate why the tuned LLM predicts well on previously unseen brands. I first empirically measure how modifying various components of the LLM affects prediction accuracy. I focus on two parts of the LLM: its ability to extract useful features from the text and its subjective prior on the mapping from the features to the outcome (like δ_{jt}). I find that adapting the features and the outcome mapping from the values initialized at a pre-trained LLM improve performance, and doing both further improves performance. However, the manner in which the parameters of the LLM are adapted are critical. Updating the LLM's parameters using standard gradient descent methods significantly harms performance. I give a theoretical justification for why restricting the rank of the parameter updates avoids these issues and empirically confirm that it does.

Finally, having established my method works well for prediction, I show how to use it for optimization. For a text description of a hypothetical new brand, I show how the

internal activations of the LLM reveal what parts of the text are driving the predicted preferences. Relying on recent advancements in sparse autoencoders, I project the internal activations of the LLM into an interpretable feature space. I measure the importance of each feature by the change in predicted δ_{jt} from deactivating it, which suggests how the brand description might be modified to increase the predicted utility.

This paper contributes to a literature on incorporating additional data into demand estimation. It complements previous work on modeling the similarity (covariance in utility) between existing products with extra data (Netzer et al. (2012), Compiani et al. (2024)). Here the focus is on predicting the mean utilities of new products. It also offers an alternative to extracting features from unstructured data and adding them to the observed attributes X_j. For example, additional product attributes may be imputed (Nevo (2001)), elicited (Dotson et al. (2019), Bell and Dotson (2022)), or discovered from auxiliary tasks (Sisodia et al. (2024)). I do not explicitly extract features but handle the raw unstructured data directly with an LLM. This allows for highly nonlinear interactions between the implicit product attributes and preference heterogeneity. Furthermore, implicit handling of product attributes contrasts with predicting demand for an entrant with a set of attributes spanned by those in the market (Li and Chintagunta (2022)) or incorporating a conjoint-based estimate of consumer preferences for a new attribute (Ellickson et al. (2019)).

The method developed in this paper also offers a solution to the well-documented problem of predicting preferences when brands change or combine (Park et al. (1996), Geylani et al. (2008), Van der Lans et al. (2014), Kim et al. (2020)). Since my framework is able to make good predictions for new brands based on cross-sectional data, it becomes possible to answer substantive marketing questions like valuing brand collaborations in a broader set of product categories and with much milder data requirements.

The remainder of this paper is as follows: Section 2 describes my method in more detail, and Section 3 evaluates its predictive performance, identifies target markets, and solves for optimal prices. Section 4 investigates the factors affecting the performance of

my method, first empirically and then theoretically. Section 5 illustrates how to apply my method to position a new brand and discover interpretable mechanisms driving consumer preferences. Section 6 concludes.

2 General Framework

I describe in detail each of the steps of my method, which I repeat below:

1. Estimate δ_{jt} for existing brands and markets with standard methods.
2. Collect text descriptions of existing brands and markets.
3. Fine-tune LLM to predict the δ_{jt}'s from Step 1 given text from Step 2.
4. Use the tuned LLM to predict δ_{jt} out of sample on new brands or markets.

2.1 Estimate δ_{jt} for existing brands with standard methods

To demonstrate and evaluate my method, I use data from the energy drinks product category as brands have a strong effect on consumer preferences in this category. Most of the incumbent brands like Red Bull and Monster have similar positioning around maximizing physical and mental performance. In recent years, however, a differently positioned brand called Celsius has been growing. It pitches itself as good for metabolism and health. I additionally observe that different customers tend to buy each of these brands in my data. The customer base for traditional energy drinks is predominantly male but much less so for Celsius, indicating a previously under-served market.

This setting is particularly challenging for existing methods because the new brand is different from existing ones in ways that are not easily captured by attributes. To evaluate the ability of various methods to predict consumer preferences for an entrant product, I examine Celsius' major retail expansion in 2022. This was facilitated by a partnership

with PepsiCo that granted access to mainstream retailers through PepsiCo's direct-store delivery (DSD) network. After estimating a demand model, I estimate a predictive model of brand preferences only using markets where Celsius was not available. I then evaluate various approaches in predicting preferences for Celsius in markets where it is available.

I obtain choice data from Numerator in the energy drinks product category. The data consists of shopping trip diaries (quantities and prices of items) across multiple purchase channels (different types of stores, online and offline). My sample consists of purchases from 2022 and contains 465,000 individuals making 3.2 million shopping trips and purchasing 6.2 million items. I define markets at the zip code level to balance granularity and sample size, and I aggregate purchases to obtain market shares for each brand.

Within the energy drink product category, suppose there are J products (I will use product and brand interchangeably) sold in T markets. An individual i's latent utility for product j is u_{ijt}, and they choose the product j that maximizes this. It is common to decompose this as

$$u_{ijt} = \delta_{jt} + \varepsilon_{ijt},$$

where δ_{jt} is the mean utility for product j within market t, and ε_{ijt} are idiosyncratic shocks.

I estimate δ_{jt} from aggregated choice data using standard methods. As a first pass, I use a simple specification of the latent utility that makes this step straightforward (Berry (1994)):

$$\hat{\delta}_{jt} = \log(s_{jt}/s_{0t}).$$

s_{jt} is the market share for brand j in market t, and s_0 is the market share of the outside option. If a more complex model that accounted for unobserved heterogeneity using random coefficients (e.g. Berry et al. (1995)) were used instead, the estimates of δ_{jt} from that model would be used in the subsequent steps. Alternatively, δ_{jt} could be estimated from a stated preference survey. All that matters is that δ_{jt} is estimated somehow from the data.

The goal is to predict δ_{jt} for new brands and markets based on the δ_{jt}'s for existing

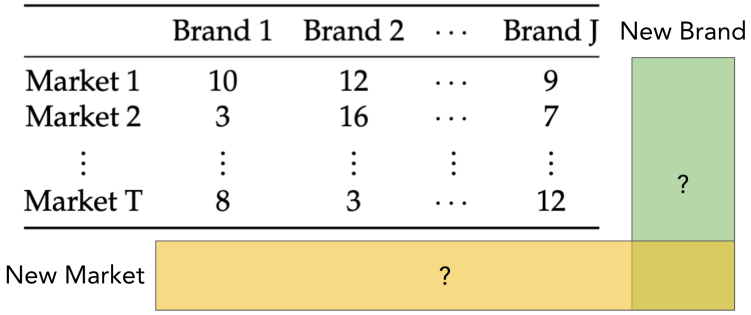


Figure 1: Arrange the estimates of δ_{jt} for existing brands and markets into a matrix. The goal is to extrapolate and predict δ_{jt} for *new* brands and markets (colored regions).

brands and markets. This is illustrated in Figure 1.

2.2 Collect text descriptions of brands and markets

The goal in this section is to gather text data that is informative about unobserved product attributes. Here I focus on text related to brands, as consumer preferences for brands were shown to explain 40% of the geographical variation in market shares across 238 product categories (Bronnenberg et al. (2012)). I also chose a product category where brand effects are strong. Other types of text data that might be useful here would be if the product category has attributes that are hard to quantify, e.g. in books, movies, or TV shows.

I gather text data that describe the energy drink brands in my data and the people that live in each zip code. This text data can be collected by assembling marketing materials from advertisements and a brand's website then manually coding features or writing a summary. Inspired by results showing that LLM-generated information is helpful for aligning LLMs (Wang et al. (2022), Zelikman et al. (2022)), I generate these text descriptions using GPT-4, the latest and most powerful language model available from OpenAI. Later, I empirically verify that the text descriptions contain useful information for predicting brand preferences out-of-sample. The specific prompt used to generate brand descriptions along with two example responses is:

System Instructions: You provide information that is relevant for market research.

Query: In 80 words or less, describe the positioning of the {`brand`} energy drink brand.

Example Response 1: Red Bull is positioned as a premium energy drink that enhances mental and physical performance. Targeted primarily at young adults, it's marketed as a vital tool for energetic lifestyles, enhancing endurance, concentration, reaction speed, and mood. Famous for its slogan "Red Bull gives you wings," the brand heavily invests in marketing through extreme sports, music festivals, and youth culture events, creating a strong, adventurous brand image that resonates globally with those seeking excitement and a performance edge.

Example Response 2: Celsius energy drink positions itself as a healthier beverage option aimed at active and health-conscious consumers. It differentiates from other energy drinks by highlighting its clinically proven formula that boosts metabolism, enhances energy, and helps burn body fat when combined with exercise. Celsius markets its products as containing no artificial preservatives or flavors, and being packed with vitamins and natural ingredients, appealing to those who seek a clean and functional energy boost for their active lifestyles.

Notice the distinct positioning of the Red Bull and Celsius brands. I manually inspected the generated descriptions for consistency with marketing materials.

Preference heterogeneity across markets is usually captured using demographic variables. As was done for brands, I tried generating text descriptions for zip codes. In the zip codes I have lived in, I found the descriptions consistent with my lived experience. Text descriptions of markets could capture more nuanced information than demographics and are easily interpretable. To the best of my knowledge, this paper is the first instance of using text descriptions of markets in demand estimation. If these text descriptions turn out to capture preference heterogeneity well, this would make the analysis of preference heterogeneity in aggregate demand models much cheaper than before, potentially replacing expensive proprietary data. The specific prompt used to generate zip code descriptions along with an example response is:

System Instructions: You provide information that is relevant for market research.

Query: In 80 words or less, describe the zip code {`zip_code`}. Talk about the types of people that live there. Do not repeat the name of the zip code in your answer.

Example Response: This suburban area is home to many families and professionals. Residents often enjoy spacious homes, good schools, and community amenities. The population includes a mix of young families, established professionals, and retirees, reflecting a range of cultural backgrounds. The area is known for its safety and family-friendly environment, with numerous parks and recreational options.

Next, I discuss how to use this text data to predict δ_{jt}.

2.3 Fine-tune LLM with the estimated δ_{jt}'s

I give the description of product j and market t to an LLM and ask it generate a prediction for δ_{jt}. If instructed, it will reply with a guess. These predictions may be bad (and probably will be), which I address below. The full prompt I use is:

System Instructions: You are a skilled analyst tasked with making forecasts for a market research firm, focusing on the competition between brands and alternative products at a local level.

Query: Given data on market shares in the energy drinks product category at the zip code level, you have estimated the average utilities for each brand within the market using a logit model. The market shares are generated by taking a softmax of the utilities. The utility of the outside option is normalized to 0.

Based on the given information, predict the utility of the described brand. Give a numerical response only, with no explanation.

Here is a description of the zip code: {`zip_code_description`}

Suppose that the following brands and utilities are known: {`known_deltas`}

Predict the average utility in this market for the brand with the following description: {`brand_description`}

Example response: 2.84

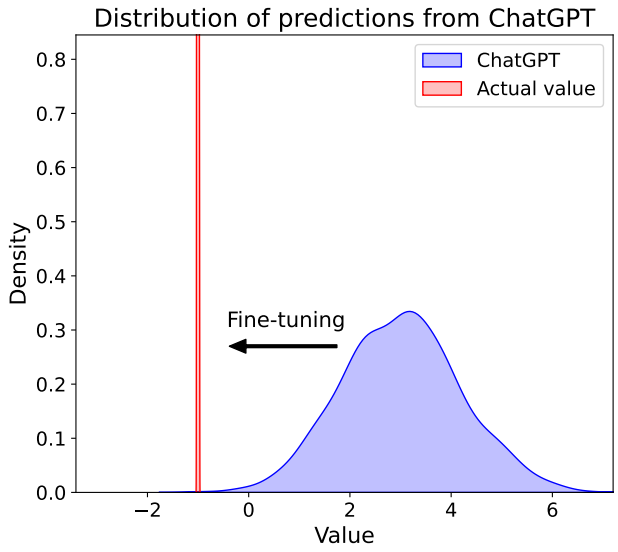


Figure 2: Repeatedly asking an LLM (e.g. ChatGPT, Mistral, Llama, etc.) to make a prediction will give a distribution of responses (blue) that may be far from the correct value of δ_{jt} (red). Updating the parameters of the LLM to maximize the likelihood of the correct value will move the blue distribution to the red.

If the above prompt is run repeatedly, it will return a distribution of responses. This is because language models are probabilistic. As depicted in Figure 2, this distribution might be far from the "correct" value of δ_{jt} that was estimated in Section 2.1.

Let the conditional probability distribution defined by the LLM be $G_\theta(\cdot|x_{jt})$. This gives the distribution over the next word given the input text x_{jt} that describes product j and market t, and θ denotes the parameters of the LLM. I update the parameters of the LLM to increase the probability mass the language model puts on the correct value of δ_{jt}, i.e. I change θ to increase $G_\theta(\delta|x_{jt})$. Formally, I fine-tune the LLM to maximize the log likelihood of the δ_{jt}'s from Step 1 given the text descriptions from Step 2:

$$\max_\theta E\left[\log G_\theta(\delta_{jt}|x_{jt})\right]$$

G_θ is a very flexible model with billions of parameters. To prevent overfitting, I only allow the parameters to change in a small neighborhood. This works well if the initialization of the parameters is good. In math, fine-tuning updates the initial parameters θ_0

to $\theta = \theta_0 + \Delta\theta$. θ_0 represents asking an untuned LLM to make a prediction for δ_{jt}, while $\theta + \theta_0$ is the version that has been tuned on the structural model's estimates and text data. I restrict $\Delta\theta$ by doing early-stopping, as is standard practice, and imposing a low-rank structure on the parameter updates (Hu et al. (2021)).

2.4 Use tuned LLM to predict δ_{jt} for new brands or markets

Finally, I use the tuned LLM to make predictions on new brands and markets. The assumption here is that the mapping from text to δ_{jt} is the same between existing brands and new brands. It treats prices as endogenous and predicts utilities assuming the same price-setting process is occurring for existing brands. If this assumption is undesirable, the extension to integrating instrumental variables estimates of price effects in Section 3.4 offers a solution.

3 Evaluation

I evaluate how well various methods might do on new data by holding out existing data. As shown in Table 1, I arrange δ_{jt} estimates for existing brands into a matrix. I partition markets by whether the entrant was available or not and train a predictive model for δ_{jt} from in markets where the entrant is *not* available. I evaluate performance in markets where the entrant is available. I denote predicting δ_{jt} for existing brands as "validation" and for the entrant as "test". The training set consists of markets that pre-date the introduction of the entrant. The validation and test set consists of markets that are post-launch of the entrant.

The validation set is useful for tuning hyperparameters and giving some sense of how the model will perform on data it has not seen while training. Looking at predictive performance on the validation set as opposed to the training set prevents being mislead by the model simply memorizing the data in the training set, which would perform quite

		Brand	
		Incumbents	Entrant
Market	Pre-launch	Train	✗
	Post-launch	Validation	Test

Table 1: Data splitting for prediction into train, validation, and test sets. Obtain δ_{jt} estimates obtained from a structural model and arrange into a matrix. Partition markets by whether the entrant was available or not. Predict δ_{jt} from explanatory variables in markets where the entrant is *not* available ("training" set). Evaluate performance in markets where entrant is available, both for existing brands ("validation" set) as well as for the entrant ("test" set). Predictive performance on the test set serves as a proxy for how our models would perform in the new product introduction setting.

poorly out of sample. The validation set is different from the test set in that any specific δ_{jt} in the validation set will have the brand j appear in the training set for a different t. In the test set, each brand j never appears in the training set. We should expect achieving strong performance in the test set to be more difficult.

The distinction between the validation and test sets is important. An example of predicting on the validation set is predicting utility of Red Bull in Boston given utility of Red Bull in Chicago. In contrast, the test set is like trying to predict utility for Celsius if it has never been seen before. If helpful, test set prediction can be viewed as a kind of cold-start problem in recommender systems. This connection is discussed further in Appendix B.

I compare the performance of my method in both the validation and test sets against several natural alternatives:

- logit model using observed product attributes (i.e. not using text data)
- untuned LLM (asking a general-purpose LLM to guess)
- standard machine learning methods based on text embeddings

3.1 Background on text embeddings

Before showing the results, I give some important background on existing methods for quantifying text and discuss intuition for why they might fail at making predictions for

previously unseen brands.

In typical practice, text is quantitatively represented via embeddings, which are vectors with 500 to 3,000 dimensions (see, e.g. Devlin et al. (2018), Reimers and Gurevych (2019), Gao et al. (2021)). Our goal is to learn a mapping from these high-dimensional objects to a numerical quantity like utility, and critically, we want this mapping to be capable of making good predictions on *unseen* text.

Learning this mapping from a 3,000 dimension embedding to the utility δ_{jt} will be hard. If this mapping is learned from scratch, it cannot say anything about dimensions of the embedding where variation was not observed in the training data. It will only learn the relationship between some of the dimensions of the text embedding and the outcome, but new brands may vary on some other dimension. I illustrate this intuition with simulated data in Appendix A and formalize this argument in Section 4.

If enough variation were observed in the data, we can imagine that a sufficiently powerful model could eventually make good out-of-sample predictions. The issue is that creating a model capable of "understanding" language and performing logical reasoning requires a massive amount of variation in the training data and an enormous model to learn the correct relationships. Empirically, the requirements are about 200 billion tokens ($\sim$150 billion words) of training data and a deep neural network of around 10 billion parameters (Wei et al. (2022), Hoffmann et al. (2022)). My choice setting does not contain enough data to train a model of this scale from scratch, and even if it did, costs would be prohibitive. But if we initialize our model to a pre-trained LLM and then adapt it to our data, our model will inherit the LLM's desired properties.

I thus use an LLM as the functional form for the model of brand effects. This calls for a different perspective on the prediction problem, as depicted in Figure 3. In traditional regression models, the text inputs would be converted to embeddings, and a transformation (e.g. LASSO, random forest) would map the embeddings to the numerical output. With an LLM, I instead give the raw text as context and instruct it to *generate* a prediction.

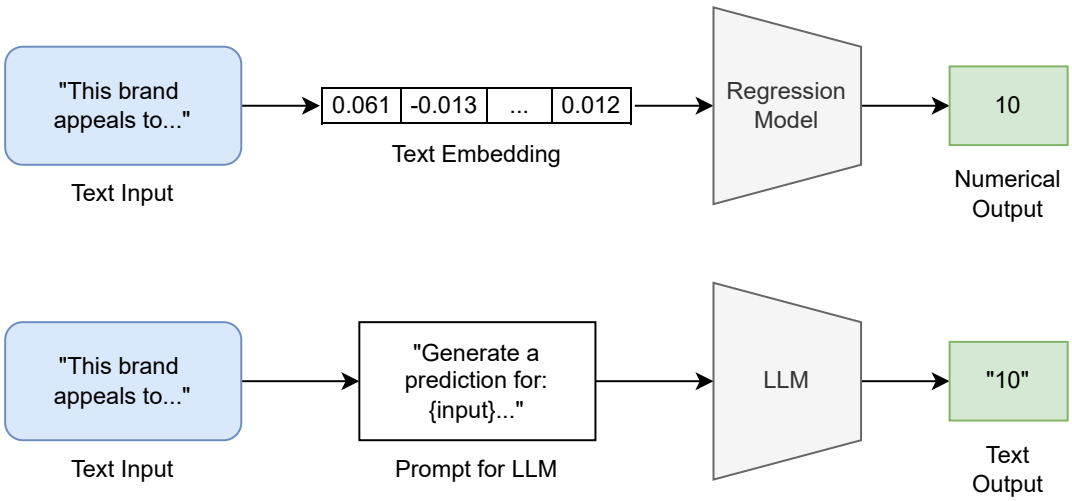


Figure 3: Two approaches to regression with text predictors. The top row is the traditional approach, which converts text to vectors using an embedding model then defines a transformation (e.g. LASSO, random forest) from the embedding to the output. The bottom row shows how an LLM can be used for regression by instructing it to generate a prediction given the input. The parameters of the LLM are then updated based on the data specific to the application.

The LLM returns text as an output (e.g. the string "5" instead of the number 5), which must then be converted back to a number. To estimate the model, I update the parameters of a pre-trained LLM via maximum likelihood (i.e. minimizing cross-entropy loss).

3.2 Prediction results

I summarize the performance of various model families in predicting the average brand-market utility δ_{jt}. Overall, four model families are evaluated:

1. "Attributes": these models predict δ_{jt} from nutritional information, which in this case are calories and caffeine content

2. "Untuned LLM": these predictions are from asking Llama 2 70B (Touvron et al. (2023)), a high quality open-source model, to make a guess given the raw text input, without any fine-tuning.

Model	Validation MSE	Test MSE
Attributes	2.17	1.64
Untuned LLM	5.24	4.09
Text Embedding	1.44	2.16
Tuned LLM	1.56	1.61

Table 2: Text data reduces error compared to models based on product attributes in the validation set. Both embeddings and the tuned LLM perform well in the validation set, but the embedding-based model does not perform well in the test set. The untuned LLM performs poorly in both cases, indicating that fine-tuning is necessary for the LLM to be useful.

3. "Text Embedding": these use embeddings of text descriptions of the brand and market descriptions. I estimate L1-regularized linear regression (LASSO), random forest (RF), and gradient boosted decision trees (GBT). These perform similarly so in this section I report the results from the lasso.

4. "Tuned LLM": these predictions are from fine-tuning Llama on our specific data

Performance is evaluated on the validation set (held out brand-market combinations containing brands that appeared somewhere in the training set) and on the test set (brands that do not appear at all in the training set). Performance on the test set serves as a proxy for how useful these models would be for decisions related to positioning an entirely new product, i.e. a cold-start setting, or alternatively, in expansion to new markets.

I first evaluate the different models by their average prediction errors. Table 2 contains the mean-squared errors of the various models, and Figure 4 plots these values. The untuned LLM does quite poorly in both the validation and test sets, indicating that fine-tuning is necessary to calibrate the LLM. Both text-based models outperform the attribute-based model in the validation set, reducing MSE by 33% relative to not using text data. On the test set, the embedding-based model does worse, and the tuned LLM helps a small amount.

Next, I compare the correlations between predicted and actual values. As shown in Figure 5, the attribute-based model has a weak negative correlation in the validation set.

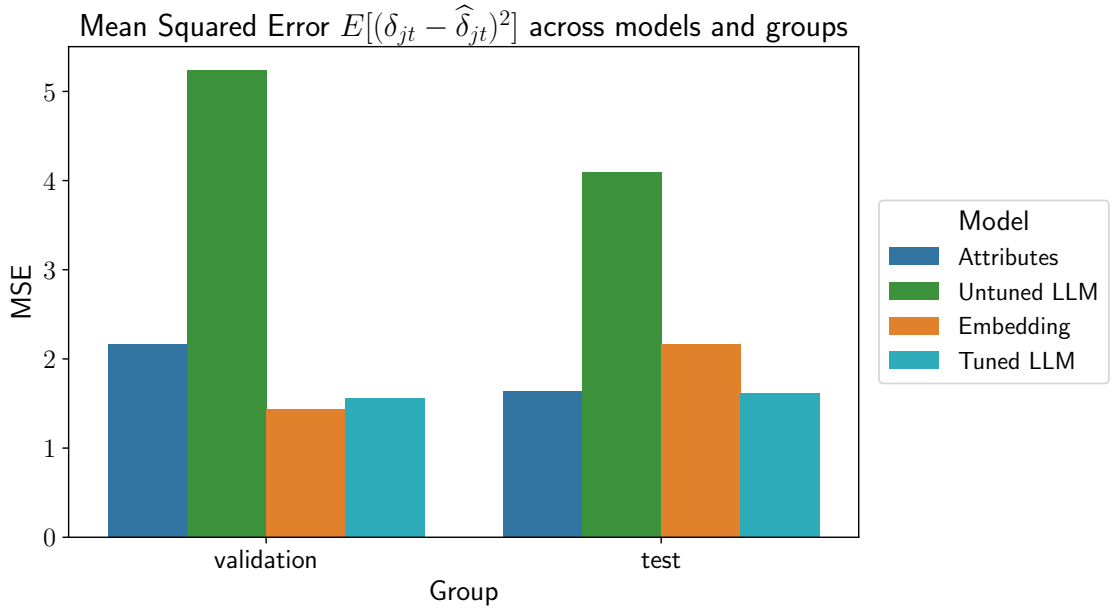


Figure 4: Plot of MSEs from Table 2. Lower is better.

While it performs reasonably at capturing mean utilities, it does not explain the across-market heterogeneity in preferences. Both the embedding and LLM-based methods attain higher correlations, with the embedding model performing the best. However, in the test set, the embedding model has a correlation of zero, so it is not useful for analyzing the heterogeneity of preferences across markets for new products. The tuned LLM does much better here, with a Pearson correlation of 0.34. Interestingly, the untuned LLM also does decently despite having high mean-squared error.

To more directly consider the problem of ranking new markets to expand into, I also compute the Spearman correlations between predicted and actual values in Figure 6. This is a measure of ordinal accuracy: it quantifies the accuracy of the ranking of markets from the predictive model. In the test set, the Spearman correlation from the attribute-based model is negative. Ranking markets based on this model will yield an ordering of markets that is negatively correlated with the true ordering. The embedding-based model has a Spearman correlation near zero, so it is not useful for ranking new markets to launch the entrant in.

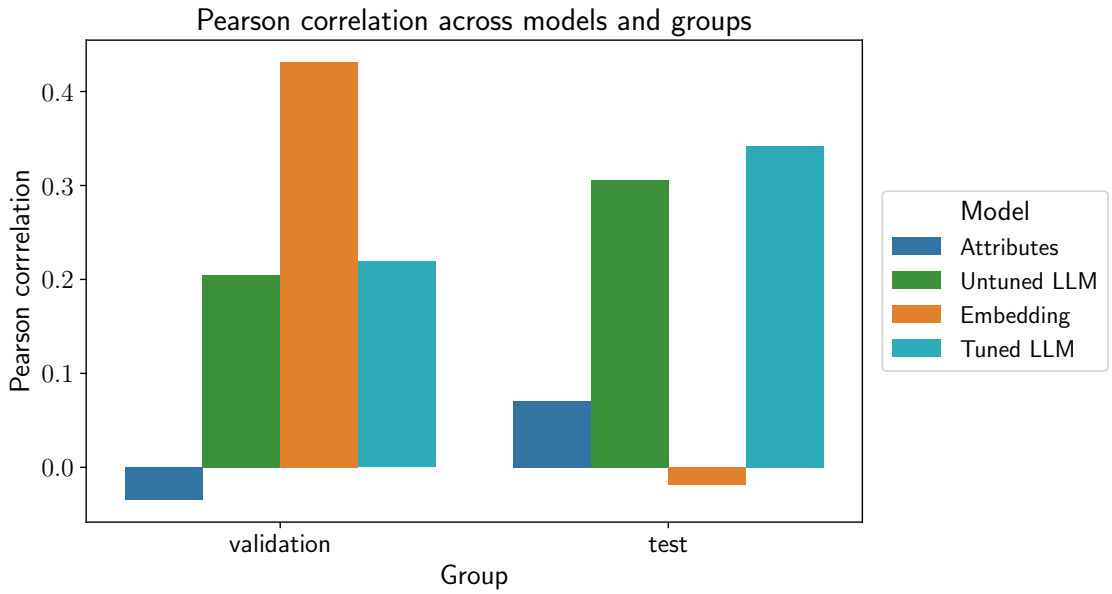


Figure 5: Pearson correlations between predicted and actual values of δ_{jt}. Higher is better All methods that use textual information give informative predictions on the validation set. For applications involving existing brands, embedding based models are the best. For new brands, embedding-based models are uninformative. Both the untuned and tuned LLMs are much better in this case.

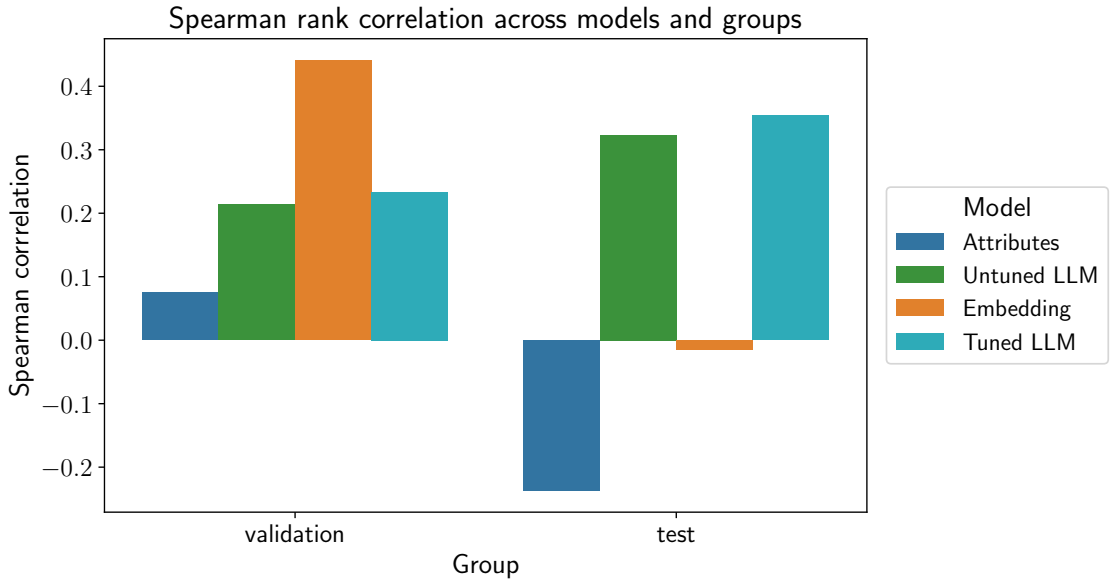


Figure 6: Spearman (rank) correlations between predicted and actual values of δ_{jt}. Higher is better. All methods that use textual information give informative predictions on the validation set. For applications involving existing brands, embedding based models are the best. For new brands, embedding-based models are uninformative, and tabular attributes alone will give an incorrect ranking of target markets. Both the untuned and tuned LLMs are much better in this case.

Finally, I compute the mutual information between predicted and actual values of δ_{jt}. Mutual information is a measure of dependence between two random variables that is more general than correlation, which measures the strength of a linear relationship. Mutual information is zero when the random variables are independent. Formally, it is the KL divergence between the joint distribution and the product of the marginals. For our setting, if the predictive model is directionally correct, higher values of mutual information are better. Otherwise, higher values are worse. In Figure 7, embedding-based models yield informative predictions in the validation set. The attribute-based method have a slightly higher mutual information but also have a negative correlation. In the test set, the embedding-based models are completely *uninformative* for predicting preferences for new products. The distributions of actual and predicted values are statistically independent. Decisions for new products thus should not rely on these models. The tuned LLM is much more informative. Again, the attribute-based model has slightly higher mutual information but has the wrong sign.

The strong performance of embedding models in the validation set suggests that they are good at interpolating but less good at extrapolating. For applications involving existing brands, embedding based models may be sufficient, but they are less suitable for applications involving new brands or changes to existing brands. Similarly, traditional demand models with brand fixed effects and product attributes work very well in the validation set but require a judgment call in the test set for the fixed effects of new brands. If we follow Berry et al. (2004) and do something like averaging the fixed effects of existing brands, these models do not make good predictions on the test set, so the text-based models are more appropriate for applications involving new brands.

I discuss preliminary results on quantifying the uncertainty of predicted δ_{jt}'s using conformal inference in Appendix C. While average prediction error is sufficient to evaluate the performance of different methods (the current focus), uncertainty quantification would be relevant for decision-making based on individual δ_{jt} predictions (a downstream

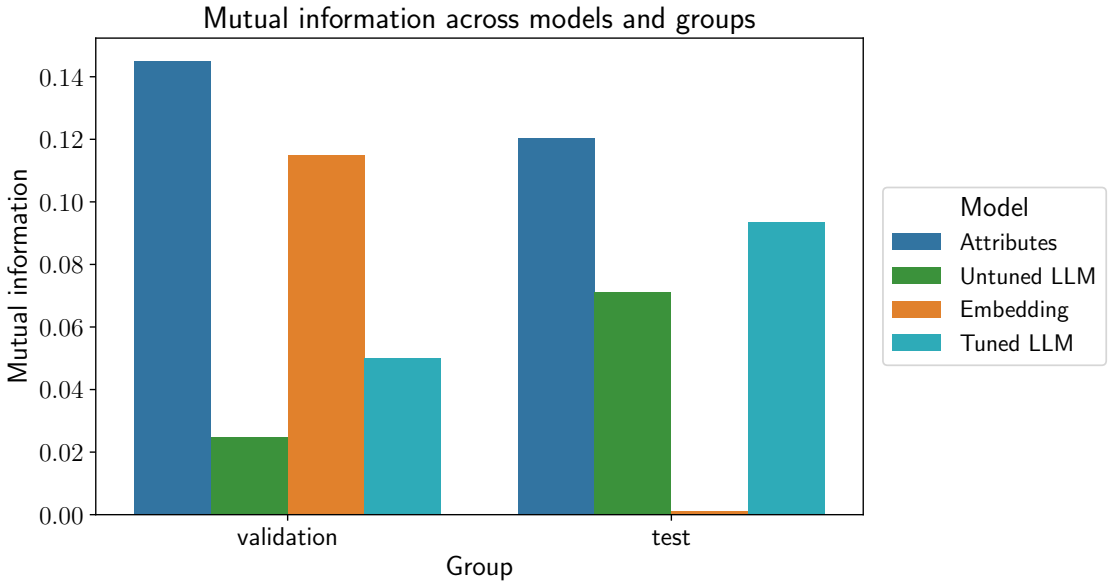


Figure 7: Mutual information between predicted and actual values of δ_{jt}. Higher is better if the direction of the relationship is correct; otherwise higher is worse. For applications involving existing brands, embedding based models are the best. For new brands, embedding-based models are completely uninformative, and tabular attributes alone are informative but have the wrong sign, as shown in the correlation plots. The tuned LLM is most informative in this case.

application).

Having established that my tuned LLM predicts well on new brands and markets, I next demonstrate two extensions to identifying target markets and optimal pricing.

3.3 Identifying target markets

Numerator only has sufficient data for computing market shares in 3,000 zip codes, but I am able to make predictions for the utilities in all 32,582 zip codes in the contiguous U.S. because my model takes a text description of the market as input and I showed it generalizes well to new markets.

For one of the test set brands, Celsius, I summarize the spatial distribution of preferences with a geographical heatmap of the predicted utilities. Figure 8, which consists of a heatmap of the predicted utilities at the 3-digit zip code level. It looks like consumers in the West and Southwest will find this product most appealing, and consumers in the

South will find it least appealing. If there are markets with high predicted utility that the brand is not sold in, those would be good candidates to explore expanding to.

This ability to generalize to new markets is practically useful. My model is able to extrapolate from high-quality data with incomplete coverage, so a smaller amount of densely collected data may be sufficient for market research applications. Another application is to introducing an existing product to new markets, e.g. for international expansion. If there is a popular product in one country, and a distributor is interested in rolling it out to a new market, consumer preferences for this product in new markets can be predicted based on text descriptions of the market.

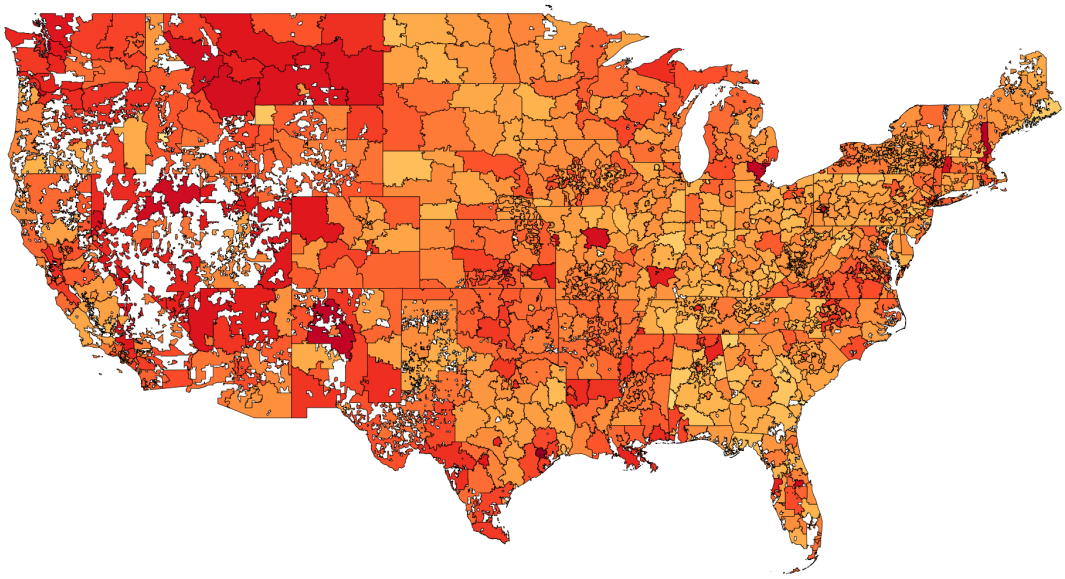


Figure 8: Predicted values of the average brand utility for one of the held out brands, Celsius, visualized in a heatmap at the 3-digit zip code level. Darker colors mean higher utility. At a quick glance, the West and Southwest appear promising, and the South less promising. This figure gives some intuition for which geographical markets might have been suitable to enter in first.

3.4 Solving for optimal prices

Counterfactuals involving changing the price of a product require causal estimates of the price effect. In structural models, the standard procedure to do this is to decompose the estimated mean utilities δ_{jt} based on a brand-specific intercept a_j and the price p_{jt}:

$$\delta_{jt} = a_j - \alpha p_{jt} + \xi_{jt}. \tag{1}$$

The problem here is that running this regression will often lead to the erroneous conclusion that higher prices lead to higher utilities. There are unobservables in ξ_{jt} that are correlated with the price, e.g. unmeasured "quality" where higher quality is correlated with higher price, which leads to biased estimates of the price coefficient. The solution is to instrument for the price. There is a long and active literature on the best way of doing so, but I use the commonly used instruments of Hausman (1994) as it is straightforward. This gives a corrected estimate for α. Alternative instruments could be used here; all that matters is α is estimated somehow.

Once α is estimated, define the quantity ϕ_{jt} that contains the non-price parts of δ_{jt}:

$$\phi_{jt} := \delta_{jt} + \alpha p_{jt}. \tag{2}$$

I compute this using α obtained from instrumental variables methods and p_{jt} from the observed price. I call ϕ_{jt} the "gross" utility, as it removes price from the "net" utility δ_{jt}.

I then train an auxiliary model to predict ϕ_{jt} from text descriptions of the brand and market. For pricing counterfactuals of a new product, one can generate the prediction for ϕ_{jt} from a text description, choose a price p and add αp to get the predicted mean utility δ_{jt} at that price, from which predicted quantities can be calculated. By multiplying the price (net of an assumed marginal cost) by the predicted quantity, we now have counterfactual predictions of the profit and can solve for the profit-maximizing price. In fact, this is suffi-

cient information to solve for the new optimal prices for *all* of the products, which allows for more realistic predictions of supply-side conduct as shown in Allenby et al. (2014).

Any other endogenous product characteristics that we know how to instrument for can be handled in a similar way. Thus estimates from causal inference, based on instruments derived from economic reasoning, can be combined with the predictive abilities of LLMs.

I estimate the price coefficient α by regressing δ_{jt} on price and brand fixed effects, with results in Table 3. I report results from OLS and from instrumenting the price with Hausman instruments (average prices in other markets). The IV estimates are obtained via GMM, as two-stage least squares is less efficient with the instruments I use. The OLS estimate is less negative than the IV estimate, which is consistent with bias from confounders. This bias is quite severe, as the IV price coefficient is 12 times the magnitude of the OLS coefficient. Still, the instruments I use are likely not perfect - the coefficients are less negative than what I would expect from other papers using scanner data. This can be remedied with better instruments. Berry et al. (1995) and Nevo (2001) discuss and compare many such options. With this estimate in hand, I construct the "gross" utilities $\phi_{jt} := \delta_{jt} + \alpha p_{jt}$, where α is the negated coefficient on price.

The results for predicting "gross" utilities ϕ_{jt} are similar to that for prediction δ_{jt}. LLM-generated predictions again significantly outperform embedding-based approaches for brands that were entirely held out of the training sample. I move these results to the appendix to simplify exposition.

I solve for the profit-maximizing price of the entrant using the estimated demand system. The market share for product j in market t implied by my demand model is

$$s_{jt} = \frac{\exp(\delta_{jt}(p))}{1 + \sum_{j'} \exp(\delta_{j't})}. \tag{3}$$

	OLS δ_{jt}	**IV-GMM** δ_{jt}
Intercept	-5.103	-4.450
	(0.035)	(0.099)
Price	-0.0224	-0.269
	(0.006)	(0.035)
Fixed Effects	Yes	Yes
Observations	22705	22705
R-squared	0.601	—

Table 3: Price coefficients obtained by regressing the net utility δ_{jt} on brand dummies and price. Since price is endogenous, I instrument for it using Hausman instruments. The IV estimate is 12x more negative than the OLS estimate, indicating that endogeneity is indeed a concern. The estimated price coefficient is used later for pricing counterfactuals.

Additionally, the utility for product j in market t at price p is

$$\delta_{jt}(p) = \phi_{jt} - \alpha p, \tag{4}$$

Profit at price p is proportional to $(p-c)s_{jt}$, where c is an assumed marginal cost. The first order condition of the profit after differentiating with respect to p is

$$\begin{aligned} \pi'(p) &= p - c - \frac{1}{\alpha(1 - s_{jt})} \\ &= p - c - \frac{1}{\alpha}\left(1 - \frac{\exp(\phi_{jt} - \alpha p)}{1 + \sum_{j'} \exp(\delta_{j't})}\right)^{-1} \end{aligned} \tag{5}$$

The second line comes from substituting Equations (3) and (4) into the first line. Notice that the price p appears twice: in the first term and in the last term.

To solve for the profit-maximizing price, I predict the utility ϕ_{jt} for the entrant from text descriptions of the product and the market. The values of $\delta_{j't}$ for the incumbents are known from estimating the structural demand model, though if we are interested in a new market, these can be predicted as well. I plug in the value of the instrumental variables estimate of the price coefficient, assume a value for the marginal cost, and solve

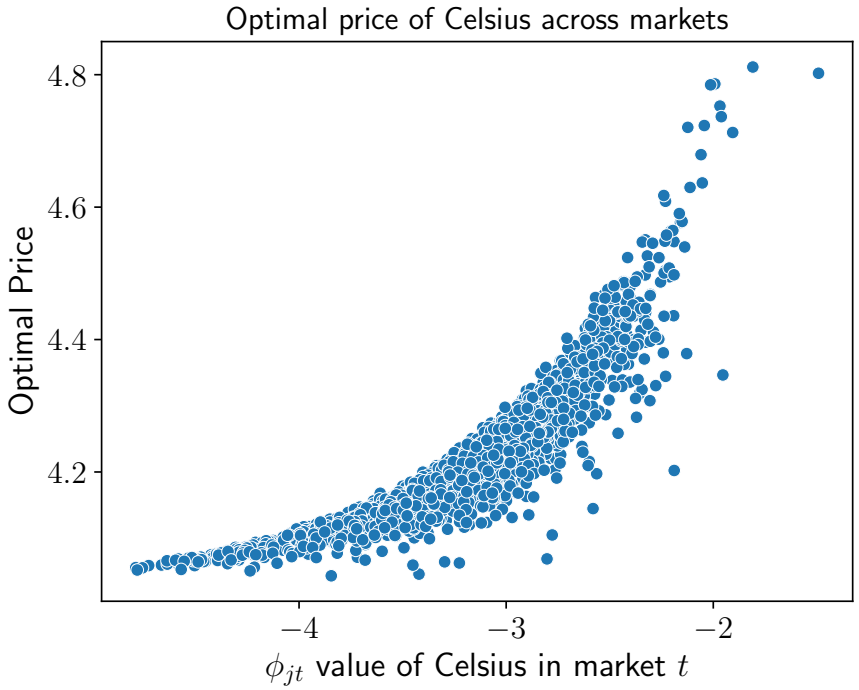


Figure 9: This figure shows the relationship between the predicted utility ϕ_{jt} for the Celsius brand and the optimal price. Each point is a zip code. The relationship is monotonic, and the vertical variation (variance in price for the same value of ϕ_{jt}) comes from the different competitive conditions in each market, i.e. differing values of the incumbent utilities $\delta_{j't}$ arising from differences in prices and preferences.

for the value of price p that makes the right hand side of Equation (5) equal to 0. Here, my estimate of the price coefficient was 0.27, and I chose to set the marginal cost to \$0.30. I check the sensitivity of the results to cost assumptions afterwards.

I predict the utility ϕ_{jt} for Celsius in each market and solve for the corresponding optimal price. Figure 9 plots the relationship between price and ϕ_{jt} across markets. The relationship is monotonic, and the vertical variation (variance in price for the same value of ϕ_{jt}) comes from the different competitive conditions in each market. The horizontal variation comes from variation in preferences. I also plot the spatial distribution of optimal prices in Figure 10.

The optimal prices are higher than would be intuitively predicted due to an inelastic estimate of the price elasticity and simplifying supply-side conduct assumptions. My instrumented price coefficient is less inelastic than is implied by running OLS without in-

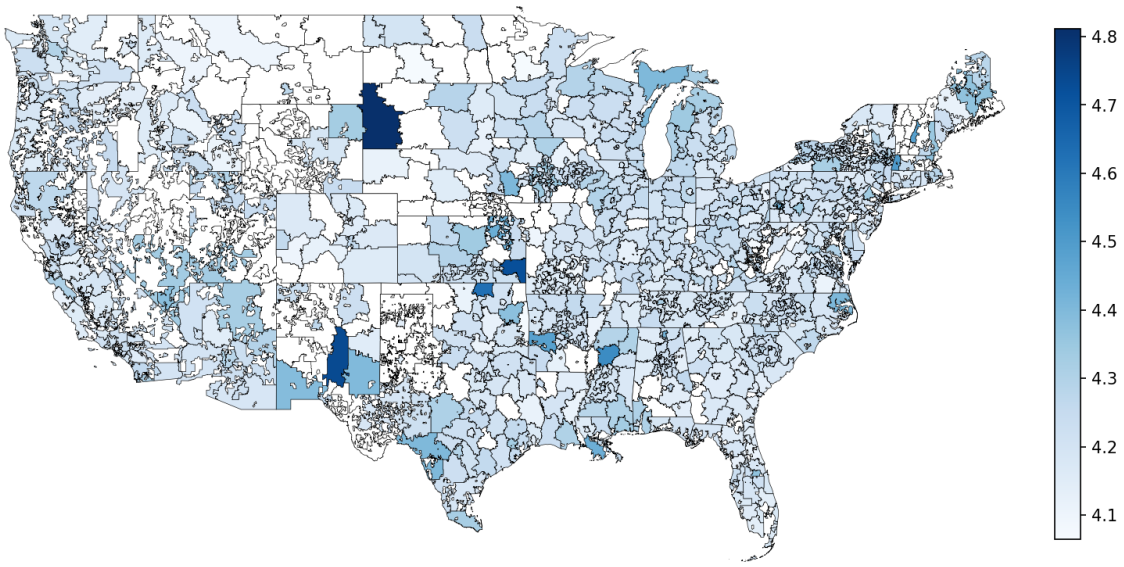


Figure 10: Spatial distribution of optimal prices for a held out brand, Celsius, averaged at the 3-digit zip code level. These prices were calculated from predictions of the utility for the product and demand parameters. Note that these prices are higher than intuitively predicted due to an inelastic estimate of the price elasticity and simplifying conduct assumptions. The white regions are where Numerator does not have sufficient coverage to compute incumbent utilities, though these can be imputed with my predictive model as well.

struments, but it is still too inelastic. Because my estimate predicts that consumers will not punish a price increase too much, the implied optimal price is higher. This can be fixed with better instruments for the price and there are plenty of examples in the literature on suggestions for such instruments (e.g. Berry et al. (1995), Nevo (2001)). Another reason the optimal prices are high is that I assume there is no competitive response by the incumbents. While this is common practice, it is not fully realistic as it describes a partial equilibrium where incumbents do not adjust prices in response to an entrant. Thus the downward price pressure from Bertrand competition is absent. Computing the full equilibrium is a more involved process and is under way. Allenby et al. (2014) shows how solving for the full equilibrium materially affects post-merger outcomes, and I expect that the same should hold for post-entry outcomes.

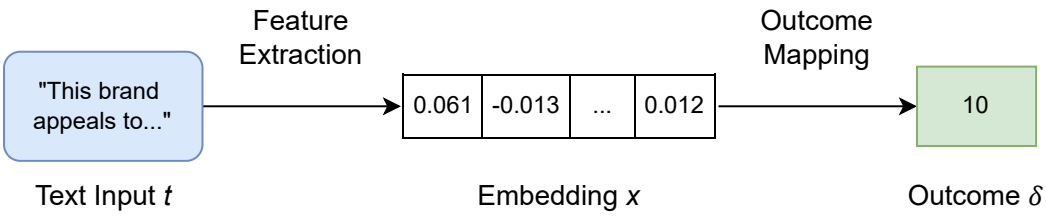


Figure 11: In regression with text inputs, raw text is turned into numerical features then mapped to the outcome. The feature extractor is taken as given from a pretrained model or adapted to the data. The outcome map is learned from scratch or adapted from a pretrained model.

4 Factors Affecting Performance

To get some sense of why the tuned LLM predicts well for unseen brands, I evaluate how changes to specific components affect prediction accuracy. I quantify the contributions from two parts of an LLM: its ability to extract useful features from text, and its subjective prior on the mapping from the features to the outcome (i.e. the utility δ_{jt}). I will show that initializing the features and the outcome map at the pretrained LLM and adapting them to the data is important, and how its done affects performance drastically.

To fix ideas, consider the simplified diagram in Figure 11. In regression with text inputs, raw text is turned into numerical features then mapped to the outcome. For example, the word count, presence of certain words, and emotional valence of input text can be recorded, then a regression model that maps these features to the outcome variable can be learned. The appeal of deep learning is that both steps are done simultaneously so that useful features are learned. Recent advances in transfer learning showed that representations learned from generic tasks (i.e. embeddings) are better than learning them from scratch (Devlin et al. (2018)). The same was later shown for the outcome mapping (Raffel et al. (2020)). I quantify how much of the value of incorporating a pretrained LLM is due to each.

I consider cases where the feature extractor is taken as given from a pretrained model or adapted to the data, and where the outcome map is learned from scratch or adapted

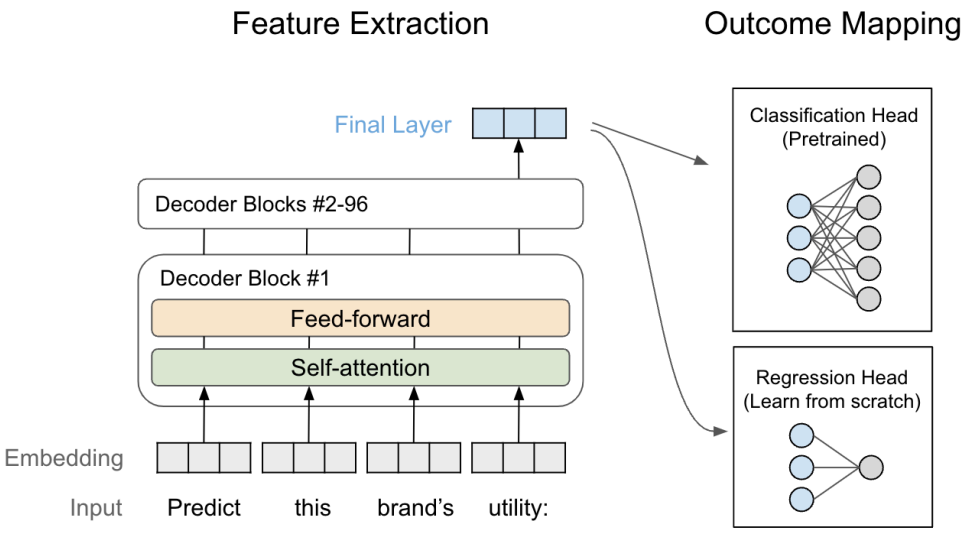


Figure 12: I take the hidden state at the last layer and last token as the text embedding to ensure differences in outcomes do not come from the quality of a different embedding model. The outcome map is either adapted from the existing classifier head, or a regression head is learned from scratch.

from a pretrained model. I will measure the predictive performance of an LLM for all 4 combinations.

To isolate the effects of each of these factors, I use a single LLM: Mistral-7B-Instruct-v0.1. I take the hidden state at the last layer and last token as the text embedding to ensure differences in outcomes do not come from the quality of a different embedding model. The outcome map is either adapted from the existing classifier head, or a regression head is learned from scratch. This is illustrated in Figure 12.

4.1 Empirical results

I compute the MSE of the predicted δ_{jt}'s for held-out brands for 4 configurations of the LLM, with results compiled in Table 4. All methods achieved similar performance on the validation set, so these results measure the difference in ability to generalize to unseen brands rather than differences in model expressivity. Learning the outcome map from scratch on frozen embeddings (top left of Table 4) performs the worst on held-out brands.

		Outcome Mapping	
		From Scratch	From LLM
Embedding	Frozen	3.136	2.718
	Adapted*	2.75	1.896

Table 4: MSE on held-out brands (test set) for various configurations of an LLM. Adapting the embedding and the LLM's prior on the outcome map are each helpful, and the combination is even more so. The way in which the embeddings are adapted affects performance drastically, which is explained in the subsequent section.

To improve set test performance, I investigate whether inheriting more information from the pretrained language model is helpful. Initialize the head of the language model to the pretrained values and fine-tuning just the head helps slightly (top right of Table 4). Next, I evaluate whether adapting the representations helps. I allow all intermediate layers of the language model to adapt to the training data, which updates the representations. This helps performance further (bottom row of 4), with a positive complementarity between adapting the representations and adapting a pretrained outcome map. That is, adapting the embedding and the LLM's prior on the outcome map are each helpful, and the combination is even more so.

However, how the embeddings are adapted affects the prediction quality dramatically. Adaptating the embeddings with standard gradient descent methods doesn't work, as shown in Table 5, which I will explain next theoretically.

		Outcome Mapping	
		From Scratch	From LLM
Embedding	Frozen	3.136	2.718
	Adapted	3.22	3.329
	Adapted*	2.75	1.896

Table 5: Adapting the embeddings with standard gradient descent (middle row) harms performance.

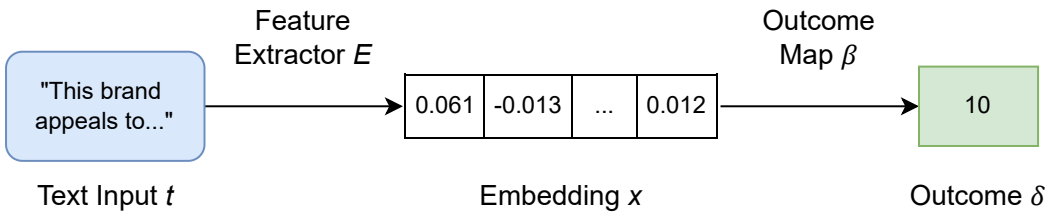


Figure 13: Text regression toy model: $\delta = \beta^\top E t$. The embedding matrix E can be frozen or adapted, the outcome map (regression coefficients) β can be learned from scratch or adapted.

4.2 Theoretical analysis

I now explain theoretically why naive adaptation of the embeddings harms predictive performance and why the alternative I used should fare better. The theory uses a toy model, as shown in Figure 13, that represents the functional form for predicting δ with a two-layer linear model:

$$\delta = \beta^\top E t.$$

t is the raw text input, which is converted into a embedding vector with a matrix E, and mapped to the outcome with regression coefficients β. E can either be frozen (taken as given) or adapted to the data, and β can be learned from scratch or adapted to the data.

My main point is an implication of a theorem in Du et al. (2018), which states that the quantity $\beta\beta^\top - EE^\top$ is preserved under gradient descent on the parameters (β, E). That is, changes to β under gradient descent must mechanically lead to changes in E. Suppose the parameters are initialized at (β_0, E_0) and that there exist optimal values (β_*, E_*). The theorem implies that if β_0 is far from β_*, which occurs when the outcome map is learned from scratch, updating β will distort E (this phenomenon was first observed in image classification models in Kumar et al. (2022)). This is undesirable if E_0 is already close to something good. I address this issue with two interventions: a better initialization of β_0 (from the pretrained language model), and restricting the changes to the E matrix. Freezing E might be too restrictive, but allowing limited changes to E might be able to

strike the right balance.

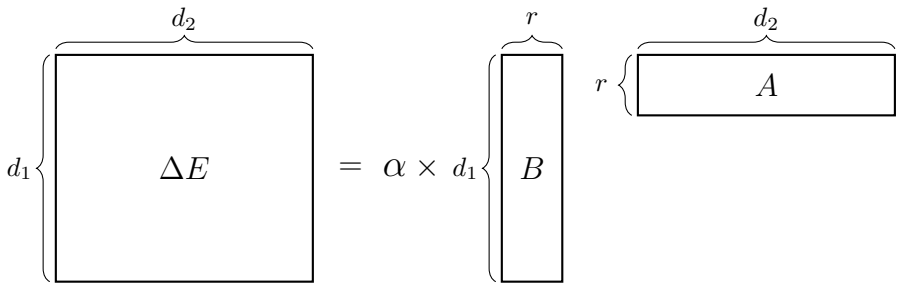
I now explain how the updates to E are restricted. Fine-tuning updates the initial embedding matrix from E_0 to $E := E_0 + \Delta E$. The claim is that the embeddings are distorted less when the rank of the ΔE matrix is restricted.

This can be shown with linear algebra. The change in the embedding for text t after fine-tuning is $\|(E_0 + \Delta E)t - E_0 t\| = \|\Delta E\ t\|$. By a singular value decomposition,

$$\|\Delta E\ t\| = \|U\Sigma V^\top t\| = \sqrt{\sum_{i=1}^{r} \sigma_i^2 (v_i^\top t)^2},$$

where r is the rank of ΔE and σ_i are the singular values of ΔE. Over different inputs t, the expected value of the change in embeddings will be proportional to $\sqrt{\sum_{i=1}^{r} \sigma_i^2}$, which is increasing in r. Thus restricting the rank of ΔE will restrict the magnitude of $\|\Delta E t\|$.

In practice, I restrict the rank of ΔE by expressing it as the product of two low-rank matrices. Suppose ΔE is a $d_1 \times d_2$ matrix. Typical values of the dimensions d_1 and d_2 in LLMs are 4,000-12,000. Instead of learning the full $d_1 \times d_2$ matrix ΔE, I factor it as:



B is a $d_1 \times r$ matrix, A is a $r \times d_2$ matrix, and α is a hyperparameter that determines the overall magnitude of the change in the parameter E. r is a hyperparameter that controls the rank of the update and is chosen to be a value $r \ll \min(d_1, d_2)$. Only B and A contain trainable parameters, and Hu et al. (2021) showed that even very small values of r (e.g. 4) performed competitively with full fine-tuning on validation-set type tasks. Given typical values of d_1 and d_2 around 4,000 to 12,000, this means cost-effective fine-tuning is possible with just 0.1% of the parameters. Visual intuition for the reduction in parameters can be

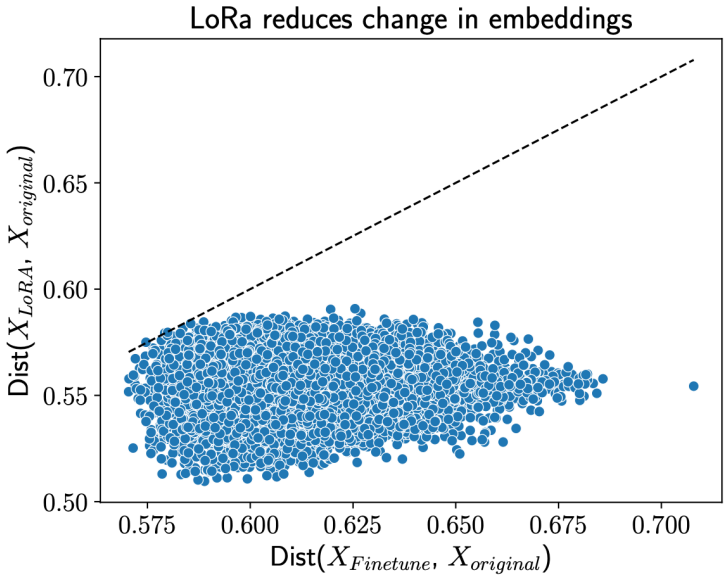


Figure 14: LoRA changes the embeddings less than full fine-tuning. For each point, I compute the distance of the embedding after fine-tuning to the original embedding (x-axis) and the distance of LoRA embedding vs the original embedding (y-axis). All points lie below the 45-degree line, and in fact lie below the horizontal line of 0.58, indicating that LoRA bounds the extent to which embeddings change.

obtained from the picture above – there are much fewer numbers in the "skinny" matrices B and A than in the full matrix ΔE.

Here my result is that low-rank adaptation (LoRA) helps improve test set performance. Imposing the low-rank structure on how the embeddings adapt improved performance in Table 4, and I empirically confirm that it also restricts the change in embeddings, as shown in Figure 14.

5 Application

Having shown my method is good at prediction of brand preferences, I now show how to use it for optimization. Consider a hypothetical entrant that is positioned distinctly from incumbent brands. I describe this brand as follows:

> ZenBoost is an energy drink positioned as a natural, calming alternative to traditional high-caffeine options. Focused on mental clarity and

stress reduction, ZenBoost features a unique blend of adaptogens, CBD, and moderate caffeine from green tea. Its branding emphasizes a holistic approach to energy, targeting professionals and students seeking sustainable focus without jitters. The packaging is eco-friendly, reflecting its commitment to sustainability and wellness, appealing to environmentally conscious consumers looking for a healthful boost in their daily routine.

For this hypothetical new brand, I show how to interpret which parts of the text description are driving preferences and present some preliminary results on how this information can be used to optimize brand positioning. Lastly, I show how to calculate perceptual maps ("brand maps") from my model. These maps visualize the nature of the competition between the entrant and incumbents.

5.1 Positioning

I investigate which parts of the text description affect the predicted utility. A benefit of generating predictions from an LLM is the ability to discover interpretable mechanisms. Similar to neuroscience imaging studies, we can investigate how patterns of neural activity within an LLM mediate the relationship between the input stimulus, e.g. the text description of a brand, and some outcome, e.g. the predicted utility δ_{jt}. Unlike in neuroscience, it is feasible to modify the activations of the neurons and observe the changes in outcome, which provides stronger mechanistic evidence than correlations alone. Suppose we identified neurons that fire when input text has a certain emotional valence and wanted to conclude that emotional valence mediates outcomes. We can suppress or amplify those neurons and test whether the outcome changes as predicted.

I conduct this exercise with the ZenBoost brand description. I fine-tuned an open-source LLM (Gemma-2-2B from DeepMind), set the residual stream activation to 0 at each layer of the LLM for each token in the text description and measured the resulting change in δ_{jt}. As shown in Figures 15 and 16, some words matter, and layer 21 of the LLM matters a lot. Since embeddings are obtained from the activations of the last layer of an LLM,

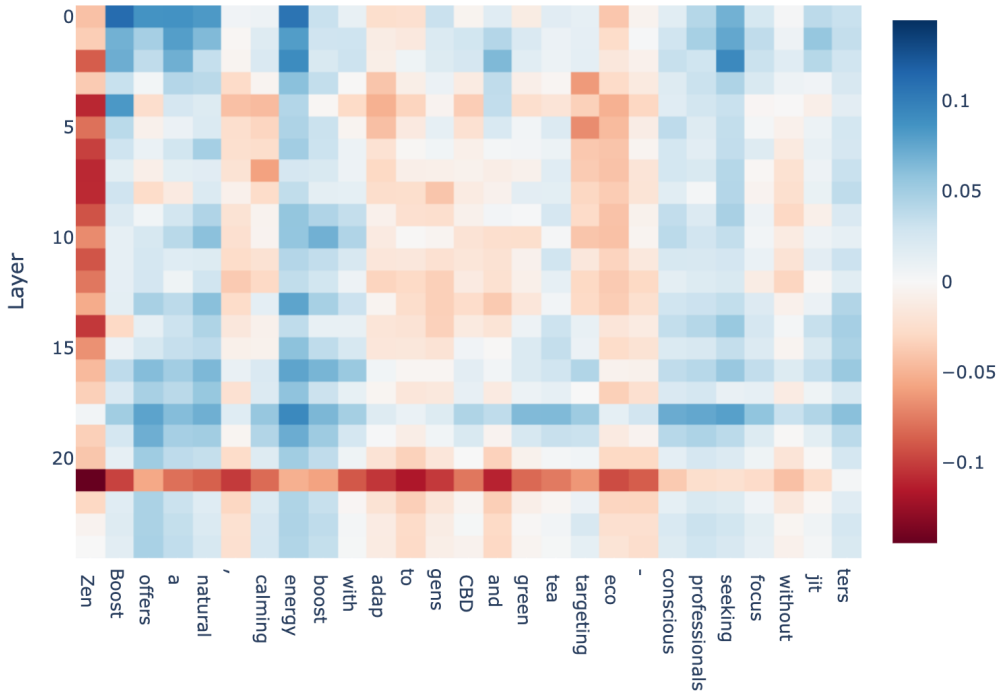


Figure 15: Ablating the residual stream at each layer of a fine-tuned LLM at each token in the brand description, which has been shortened to fit on the page. The color is the resulting change in the predicted δ_{jt}. Layer 21 of the LLM matters a lot.

this analysis would not have been possible with an embedding-based model. These kinds of figures should be useful for exploratory analyses that seek to identify factors within the brand or customer description that drive preferences. This can inform decisions on repositioning the brand as a whole or identifying the kinds of customers that the brand should be targeted toward.

Deactivating an entire layer of the LLM is a very coarse operation, so I now investigate layer 21 in finer detail. Unfortunately, the individual neurons of the LLM are difficult to interpret: for a given type of input text, many neurons fire, and a given neuron will fire for many types of input text (Bricken et al. (2023)). However, the neuron activations can be mapped into an interpretable space using a sparse autoencoder. The autoencoder performs a version of factor analysis where the number of factors is *larger* than the number of dimensions. Imposing a sparsity penalty on the factor loadings leads to the learned factors, hereafter referred to as "features", being interpretable – the factors only load on

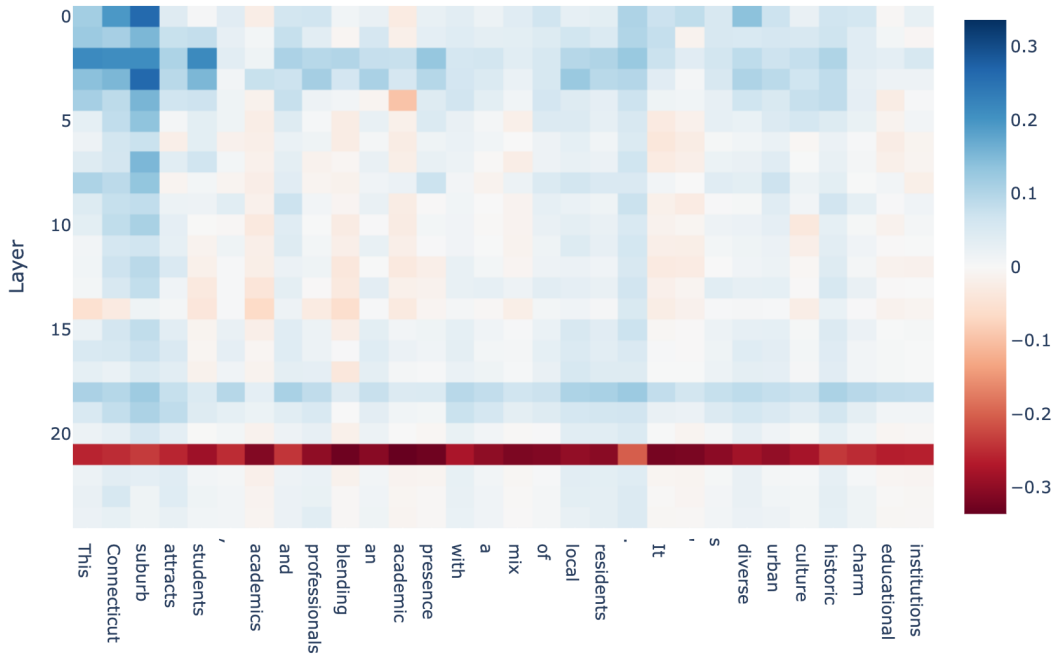


Figure 16: Ablating the residual stream at each layer of a fine-tuned LLM at each token in a **market** description. The color is the resulting change in the predicted δ_{jt}. Layer 21 of the LLM and the words "suburb" and "student" matter a lot.

very specific kinds of input text (Cunningham et al. (2023)). Additionally, changing the loadings on features changes the generated text in the expected ways (Templeton et al. (2024)).

Formally, let $\mathbf{h}$ be intermediate values of LLM at a layer. $\mathbf{h}$ is not interpretable, so we project it into a "feature space":

$$\hat{\mathbf{h}} = \sum_{k=1}^{K} c_k(\mathbf{h})\, \mathbf{f}_k$$

The result from Cunningham et al. (2023) is that imposing sparsity on the coefficients $c_k(\mathbf{h})$ leads to interpretable features $\mathbf{f}_k$. Or in math: $\min_{\mathbf{f},\mathbf{c}(\cdot)} \|\mathbf{h}-\hat{\mathbf{h}}\|_2^2+\lambda\|\mathbf{c}(\mathbf{h})\|_1$ leads to interpretable $\mathbf{f}_k$. This sparse autoencoder must be trained ahead of time (i.e. it is pretrained). While doing this, the learned features can also be named by using another LLM to summarize the kinds of text that activate it.

I compute the feature activations of the brand description by attaching a pretrained sparse autoencoder (Lieberum et al. (2024)) to the internal activations of my LLM, as il-

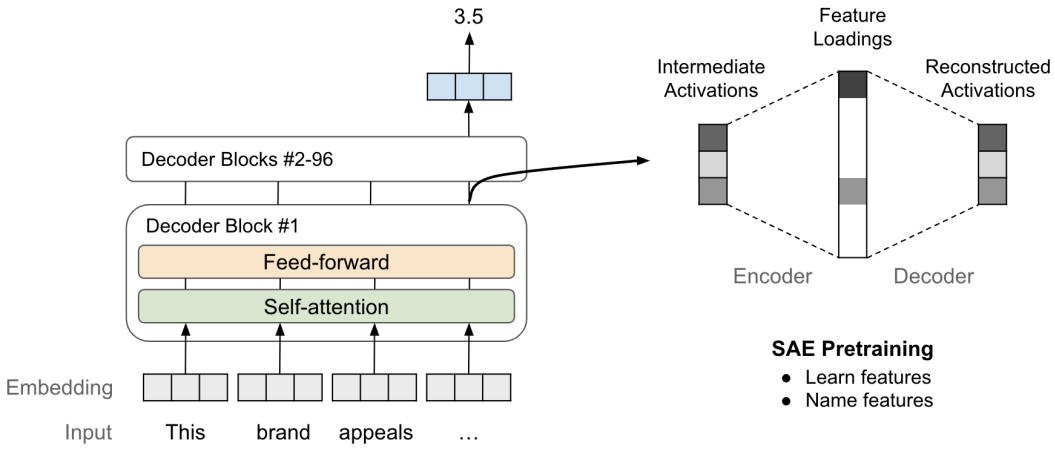


Figure 17: Attaching a sparse autoencoder (SAE) to an LLM enables interpretation of intermediate values. The SAE must be separately trained beforehand to learn the features and name them.

lustrated in Figure 17. These activations are plotted in Figure 18. 92.5% of the features have zero activation, and the top activating feature is "phrases related to nutritional supplements and their effects".

The feature activations only describe the input text. To assess the impact of the features on the predicted utility, I set the activation of each feature to 0 and measure the effect on the output. This requires several steps: I attach the autoencoder to the LLM, apply the encoder to the LLM's activations, set the coefficient on one of the features to 0, run the decoder on the modified values, then replace the LLM's activations by the reconstructed activations from the decoder. The change in the value of the δ_{jt} can be viewed as the predicted causal effect of ablating the feature. Discovered features with large effects on the outcome are: "phrases related to health and wellness", "references to plants and plant-based topics", and "terms related to medical or scientific findings".

Given estimates of which features matter, we can complete the feedback loop and apply it to modifying the description of the brand. This is done similarly as in the previous part, but instead of setting the activation of a feature to 0, its value is increased. I illustrate this procedure in Figure 19. This method offers an interpretable way of optimizing brand positioning. It is also a natural point for an analyst to intervene and impose their domain

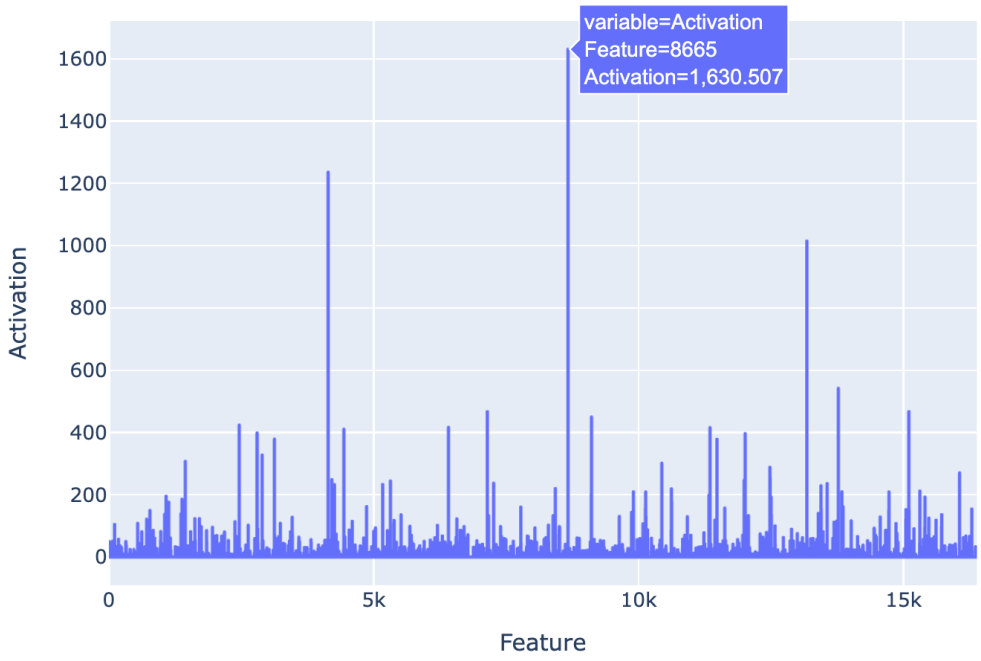


Figure 18: These feature activations of the ZenBoost brand description are computed by attaching a sparse autoencoder to the internal activations of my LLM. 92.5% of the features have zero activation, and the top activating feature is "phrases related to nutritional supplements and their effects".

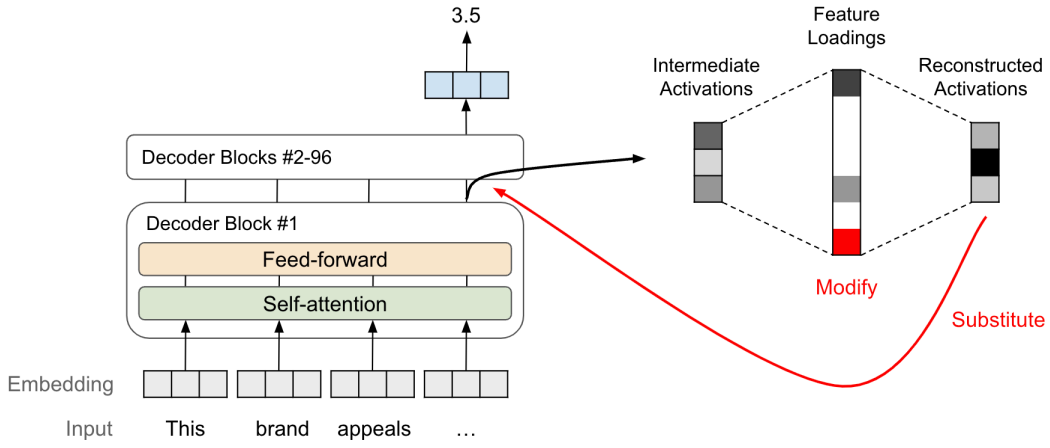


Figure 19: Amplification of features with an SAE. The intermediate values of an LLM are projected into the interpretable feature space, modified as desired, then reconstructed. The reconstructed values replace the original values, and the new output is computed.

knowledge. If this analysis reveals that the LLM is loading on spurious or undesirable features, the entire LLM fine-tuning loop can be re-run with the constraint that certain features do not activate. In Table 6, I show some results from boosting the "plant-based" and "health/wellness" features deemed to be important in the previous step. Amplifying the feature too much degrades the fluency and relevance of the generated text, but this is due to the relatively small 2 billion parameter being used here. This is less of an issue with larger models, and results will be updated once those analyses are complete.

A similar analysis can be performed by applying the sparse autoencoder to the market description, with the goal of identifying customer types with high match values with the brand. Overall, this section illustrates the additional interpretation and control that LLM's make possible beyond embedding-based models.

5.2 Brand Map

To succinctly summarize how this brand competes with existing brands, I obtain predictions of the net utility δ_{jt} for this brand across all markets and compare it against the δ's for the incumbent brands. I create a visual representation similar to the spatial mapping literature (Elrod (1988), Chintagunta (1994)). Representing each brand by its full vector of utilities δ_{jt} across markets, I apply a dimensionality reduction technique, t-SNE (Van der Maaten and Hinton (2008)), on this vector, as shown in Figure 20. Intuitively, it makes sense that the new brand is closer to Celsius than most of the other brands.

Feature	Steering strength	Generated text
	Original	ZenBoost offers a natural, calming energy boost with adaptogens, CBD, and green tea, targeting eco-conscious professionals seeking focus without jitters.
Plant-based	Low	ZenBoost is a plant-based, energy drink that provides a natural, calming energy boost without the jitters
	Medium	I'm a plant-based, non-GMO, caffeine-free, and CBD-based energy drink with no crash and no jitters. I'm the new way to get your life back on track.
	High	Life is busy. But it has to be. That's life. And with life comes stress.
Health	Low	The ZenBoost brand is a natural and effective energy drink that is designed to provide a boost of energy while also promoting relaxation.
	Medium	Our brand is about providing a more natural, more effective, and more sustainable alternative to the traditional energy drink.
	High	ZenBoost is a natural, non-stimulant, and non-dehydrating, and non-negative impact on the environment

Table 6: Modifying the brand description by boosting interpretable features in an LLM. If data reveals that a certain feature is desirable, this offers a way to operationalize that information. If the magnitude of the intervention is too large, it starts to distort the output, but this is less of an issue with larger language models. Alternatively, these features could be boosted by simply instructing the LLM to rewrite the text with emphasis on the feature.

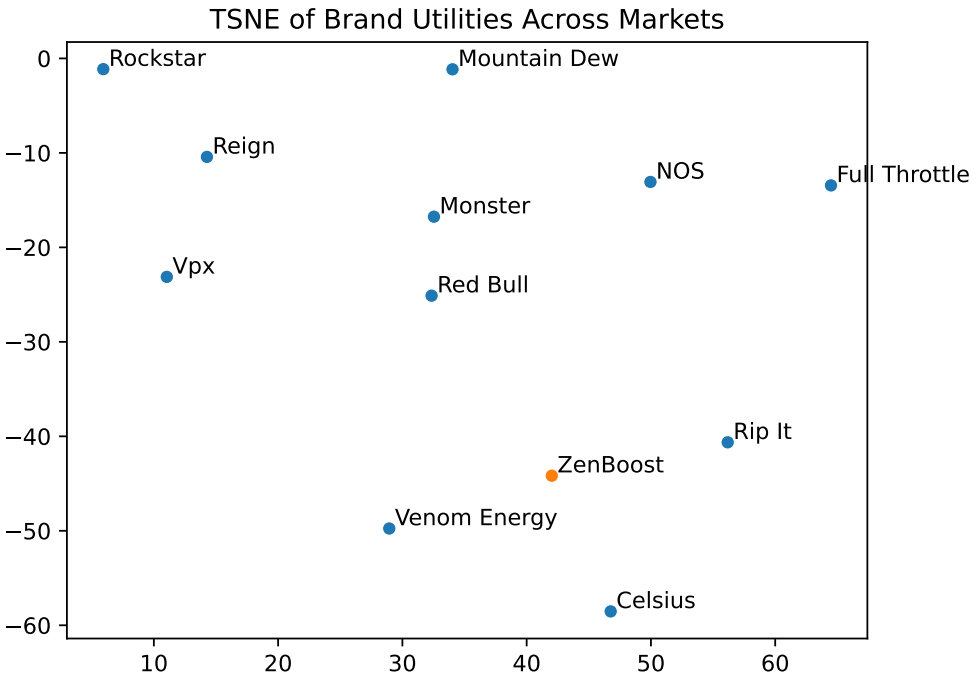


Figure 20: Applying a dimensionality reduction technique, t-SNE (Van der Maaten and Hinton (2008)), to the vector of brand utilities across markets gives a visual depiction of brand similarity. The position of the hypothetical new brand is marked in orange, and it intuitively makes sense that it would be closer to Celsius than most of the other brands.

6 Conclusion

I developed a scalable method for modeling brand preferences from text that generalizes to unseen brands and markets. The keys to the success of my method were initializing and then adapting text feature extractors and subjective priors from a pretrained LLM. Moreover, imposing a low-rank structure on the updates to the LLM's parameter was critical for my method to work.

My method gives a framework for predicting demand for new products as a function of marketing mix variables like product position, price, and place, which I demonstrated through a series of counterfactuals. With these predictions, it becomes possible to optimize these marketing mix variables. While I focused on predicting preferences for new brands, the framework is more general and can enhance any demand model where brand fixed effects play a prominent role. Additionally, compared to existing approaches of mea-

suring brand preferences that require additional psychometric data, my method is cheaper and more accessible. LLM-generated text descriptions contain enough useful information for modeling preferences, and low-rank adaptation reduces the costs of fine-tuning.

Substantively, this paper shows that informative counterfactual predictions related to positioning and pricing a new product are feasible from raw text descriptions of a brand, which is new in the literature. This was accomplished by combining data on consumers' choices with text data on brands within a framework that integrated structural models and LLMs. Whereas traditional text embedding-based models were unable to make informative predictions, carefully tuning LLMs on results from an economic model proved successful at harnessing their abilities for our specific problem.

Practically, this framework makes questions that were previously unanswerable tractable. For example, an energy drink entrant like Celsius can optimize its positioning, identify what kinds of markets would be the best match, and solve for profit-maximizing prices. An incumbent brand like Red Bull can explore how to optimally reposition, e.g. if it wants to change how it positions its brand, with this framework.

The framework can also be used to operationalize hypotheses from the behavioral literature. If lab studies showed that the emotional valence of brand positioning moderates brand preferences, LLM steering techniques (e.g. Keskar et al. (2019)) can be used to modulate the emotional valence of generated brand descriptions and test the effects of doing so. Additionally, my framework offers a complementary setting to the lab for studying how brands affect consumer perceptions (e.g. Simonin and Ruth (1998)). Given a description of a customer, my framework can be used to check whether varying the description of a brand in a specific way leads to the directional changes in utility we would expect from a lab study, and it can quantify the monetary value of such changes in perceptions.

Beyond the immediate practical applications, this framework is useful for answering policy questions. One example is that it makes a much more expressive version of merger simulations possible. One reason for the poor predictive performance of merger simula-

tion models is that they do not account for how product characteristics will change after a merger (Peters (2006)). It should be feasible to obtain expert opinions on how firms will change their product positioning from analyst reports or interviews of executives. Then an LLM can be take pre-merger product positions and these reports as input and generate a prediction for post-merger product positioning, which is sufficient information for the framework to model the new demand system.

More broadly, this paper shows how generative models enable the integration of unstructured data into choice models. It would be interesting to extend these ideas to other forms of unstructured data, such as images of advertisements or open-ended responses to interview questions asking about consumers' needs. For the latter, rich inferences of how products fulfill these needs can be calculated from choice data. Future work on these applications is underway.

References

Allenby, G. M., J. Brazell, J. R. Howell, and P. E. Rossi (2014). Valuation of patented product features. *The Journal of Law and Economics 57*(3), 629–663.

Bell, J. J. and J. P. Dotson (2022). Phantom attributes: Unpacking product perceptions. *Available at SSRN 4109569*.

Bennett, J. and S. Lanning (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, Volume 2007, pp. 35. New York.

Berry, S., J. Levinsohn, and A. Pakes (1995). Automobile prices in market equilibrium. *Econometrica 63*(4), 841–890.

Berry, S., J. Levinsohn, and A. Pakes (2004). Differentiated products demand systems from a combination of micro and macro data: The new car market. *Journal of political Economy 112*(1), 68–105.

Berry, S. T. (1994). Estimating discrete-choice models of product differentiation. *The RAND Journal of Economics 25*(2), 242–262.

Bricken, T., A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. L. Turner, C. Anil, C. Denison, A. Askell, R. Lasenby, Y. Wu, S. Kravec, N. Schiefer, T. Maxwell, N. Joseph, A. Tamkin, K. Nguyen, B. McLean, J. E. Burke, T. Hume, S. Carter, T. Henighan, and C. Olah (2023). Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread 2*.

Bronnenberg, B. J., J.-P. H. Dubé, and M. Gentzkow (2012). The evolution of brand preferences: Evidence from consumer migration. *American Economic Review 102*(6), 2472–2508.

Bronnenberg, B. J. and C. F. Mela (2004). Market roll-out and retailer adoption for new brands. *Marketing Science 23*(4), 500–518.

Chintagunta, P. K. (1994). Heterogeneous logit model implications for brand positioning. *Journal of Marketing Research 31*(2), 304–311.

Compiani, G., I. Morozov, and S. Seiler (2024). Demand estimation with text and image data. *Available at SSRN 4588941*.

Cunningham, H., A. Ewart, L. Riggs, R. Huben, and L. Sharkey (2023). Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dotson, J. P., M. A. Beltramo, E. M. Feit, and R. C. Smith (2019). Modeling the effect of images on product choices.

Du, S. S., W. Hu, and J. D. Lee (2018). Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. *Advances in neural information processing systems 31*.

Ellickson, P. B., M. J. Lovett, and B. Ranjan (2019). Product launches with new attributes: a hybrid conjoint–consumer panel technique for estimating demand. *Journal of Marketing Research 56*(5), 709–731.

Elrod, T. (1988). Choice map: Inferring a product-market map from panel data. *Marketing Science 7*(1), 21–40.

Gal, Y. and Z. Ghahramani (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR.

Gao, T., X. Yao, and D. Chen (2021). Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Gemma Team (2024). Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Geylani, T., J. J. Inman, and F. T. Hofstede (2008). Image reinforcement or impairment: The effects of co-branding on attribute uncertainty. *Marketing Science 27*(4), 730–744.

Hausman, J. A. (1994). *Valuation of new goods under perfect and imperfect competition*. National Bureau of Economic Research Cambridge, Mass., USA.

He, Z., Z. Xie, R. Jha, H. Steck, D. Liang, Y. Feng, B. P. Majumder, N. Kallus, and J. McAuley (2023). Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 720–730.

Hoffmann, J., S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, et al. (2022). Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Hu, E. J., Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Kamakura, W. A. and G. J. Russell (1993). Measuring brand value with scanner data. *International journal of Research in Marketing 10*(1), 9–22.

Keskar, N. S., B. McCann, L. R. Varshney, C. Xiong, and R. Socher (2019). Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Kim, Y., S. Misra, and B. Shapiro (2020). Valuing brand collaboration: Evidence from a natural experiment. *Available at SSRN 3335833*.

Kumar, A., A. Raghunathan, R. Jones, T. Ma, and P. Liang (2022). Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*.

Lakshminarayanan, B., A. Pritzel, and C. Blundell (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems 30*.

Lancaster, K. J. (1966). A new approach to consumer theory. *Journal of political economy 74*(2), 132–157.

Lei, J., M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association 113*(523), 1094–1111.

Li, J., M. Wang, J. Li, J. Fu, X. Shen, J. Shang, and J. McAuley (2023). Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1258–1267.

Li, W. and P. K. Chintagunta (2022). Product attributes, cross elasticities and dynamic market structure in a category with many products.

Lieberum, T., S. Rajamanoharan, A. Conmy, L. Smith, N. Sonnerat, V. Varma, J. Kramár, A. Dragan, R. Shah, and N. Nanda (2024). Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *arXiv preprint arXiv:2408.05147*.

McFadden, D. (1974). Conditional logit analysis of qualitative choice behavior. In P. Zarembka (Ed.), *Frontiers in Econometrics*, pp. 105–142. Academic Press.

Netzer, O., R. Feldman, J. Goldenberg, and M. Fresko (2012). Mine your own business: Market-structure surveillance through text mining. *Marketing Science 31*(3), 521–543.

Nevo, A. (2001). Measuring market power in the ready-to-eat cereal industry. *Econometrica 69*(2), 307–342.

Park, C. W., S. Y. Jun, and A. D. Shocker (1996). Composite branding alliances: An investigation of extension and feedback effects. *Journal of marketing research 33*(4), 453–466.

Peters, C. (2006). Evaluating the performance of merger simulation: Evidence from the us airline industry. *The Journal of law and economics 49*(2), 627–649.

Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research 21*(140), 1–67.

Reimers, N. and I. Gurevych (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl (1994). Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175–186.

Simonin, B. L. and J. A. Ruth (1998). Is a company known by the company it keeps? assessing the spillover effects of brand alliances on consumer brand attitudes. *Journal of marketing research 35*(1), 30–42.

Sisodia, A., A. Burnap, and V. Kumar (2024). Express: Generative interpretable visual design: Using disentanglement for visual conjoint analysis. *Journal of Marketing Research*, 00222437241276736.

Su, X. and T. M. Khoshgoftaar (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence 2009*(1), 421425.

Templeton, A., T. Conerly, J. Marcus, J. Lindsey, T. Bricken, B. Chen, A. Pearce, C. Citro, E. Ameisen, A. Jones, H. Cunningham, N. L. Turner, C. McDougall, M. MacDiarmid, A. Tamkin, E. Durmus, T. Hume, F. Mosconi, C. D. Freeman, T. R. Sumers, E. Rees, J. Batson, A. Jermyn, S. Carter, C. Olah, and T. Henighan (2024). *Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet*. Anthropic.

Touvron, H., L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur,

S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Van der Lans, R., B. Van den Bergh, and E. Dieleman (2014). Partner selection in brand alliances: An empirical investigation of the drivers of brand fit. *Marketing science 33*(4), 551–566.

Van der Maaten, L. and G. Hinton (2008). Visualizing data using t-sne. *Journal of machine learning research 9*(11).

Walker, J. and M. Ben-Akiva (2002). Generalized random utility model. *Mathematical social sciences 43*(3), 303–343.

Wang, Y., Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi (2022). Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Wei, J., Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus (2022). Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Zelikman, E., Y. Wu, J. Mu, and N. Goodman (2022). Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems 35*, 15476–15488.

A Example showing limitations of embeddings

Here is a simulated example that illustrates the limitations of working with text embeddings using simulated data. The purpose of this example is to have a minimally complex demonstration of issues with learning to generalize from text inputs.

Consider a setting where we have a single "ideal test market". In this market, only 1 product is available at a time, and we observe the proportion of consumers that buy the product. We model this choice data with a standard logit demand model:

$$\begin{aligned} u_{ij} &= \delta_j + \varepsilon_{ij} \\ \implies Pr(y_j = 1) &= \frac{\exp(\delta_j)}{1 + \exp(\delta_j)}. \end{aligned}$$

Let's try this where $\delta_j = \delta(\text{text}_j)$ is modeled as a function of the embedding of the text description of the product. Suppose that if a product that "appeals to those who care about quality" were available, 80% of consumers would purchase it, and if a product that "appeals to those who care about convenience" were available, 30% of consumers would purchase it. A logit demand model estimated on this data predicts similar choice outcomes for products with *opposite* meanings, as shown in Table 7. Because this specific variation was not in our data, our logit model was not able to learn that the change in the embeddings from negating the meaning should lead to a change in the outcome. Interestingly, simply asking ChatGPT to make a guess gives more plausible predictions. It already "understands" that negating the meaning of the brand description will likely change the outcome. Additional details on this example, like the estimation procedure and the prompt I used for ChatGPT, are in Appendix A.

Implementation details for this example are as follows. The embedding model is OpenAI's text-embedding-3-small, their latest and best performing text embedding model. For the logit model, the product utility is modeled as the dot product of a learned user

	Brand Description ("appeals to those who...")	Predicted Purchase Probability	
		Logit	GPT-4
In-sample	care about quality	79.2%	80%
	care about convenience	31.5%	30%
Out-of-sample	care about value	70.5%	50%
	do not care about value	64.9%	20%

Table 7: Example of challenges with out-of-sample prediction with text inputs. The top two rows are the observed data: 80% of consumers in a market will buy a product that appeals to those who care about quality, and 30% will buy a product that appeals to those who care about convenience. The bottom two rows describe products not in the training sample. A logit model over text embeddings predicts similar outcomes for brands with opposite meanings, as seen in the red numbers, because this kind of variation was not in the training sample. In contrast, asking ChatGPT to guess leads to more plausible predictions, as it "knows" that negating the meaning should change the outcome.

vector and the text embedding of the product description:

$$\delta(\text{text}_j) = X_j\beta.$$

The implied choice probabilities are $\exp(X_j\beta)/(1+\exp(X_j\beta))$. Given data simulated for 100,000 consumers, β is estimated via maximum likelihood with elastic net regularization (penalty .001, equally split between L1 and L2).

ChatGPT predictions are obtained by putting the known data into the prompt:

> **System Instructions**: You are a skilled analyst tasked with making market share forecasts for a market research firm, focusing on the competition between brands and alternative products.
>
> **Query**: Consider a scenario where there is only 1 market that has 1 product available at a time. Suppose that if a product that "appeals to those who care about quality" were available, 80% of consumers would purchase it, and if a product that "appeals to those who care about convenience" were available, 30% of consumers would purchase it. What percent of customers in the market do you estimate will buy if the available brand "`{new_brand}`"? Give a numerical response only, with no explanation
>
> **Example Response**: 20%

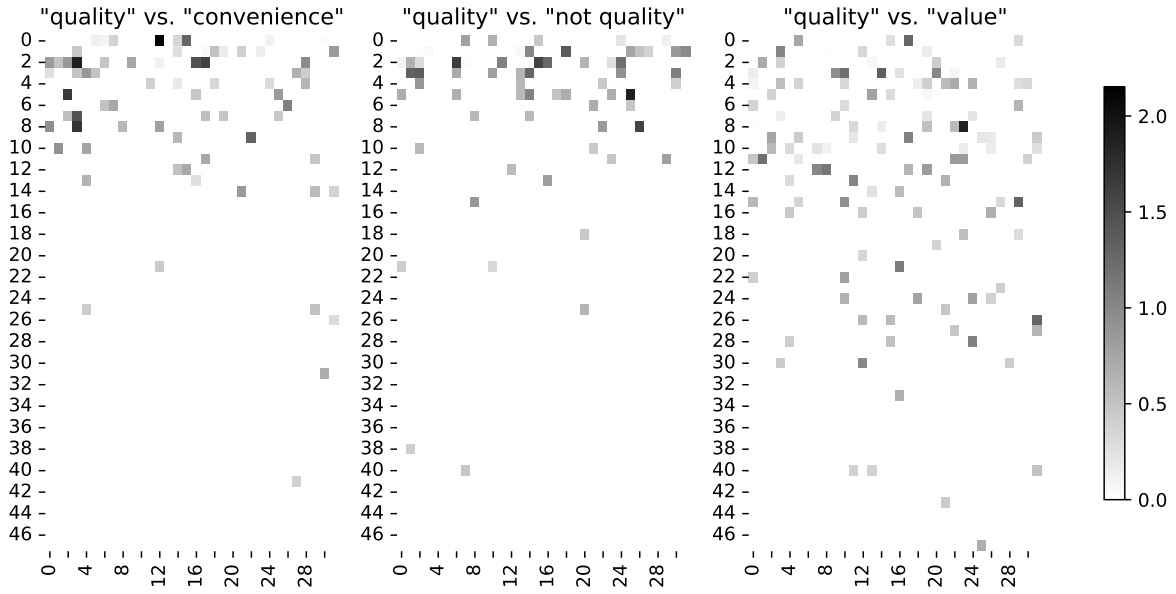


Figure 21: A heatmap of the loadings on the 1536-dimensional embedding of a brand description if only two products are observed in the training data. For visualization purposes, the 1536-dimensional vector has been reshaped into a 48 by 32 array, and the color of each entry is the magnitude of the coefficient from the logit model. Note that different dimensions of the embedding matter depending on what is observed. This figure provides intuition for why linear models over embeddings will struggle to make good predictions out of sample.

Additionally, we can visualize what dimensions of the embedding the logit model loads on depending on the variation that is observed. The plots in Figure 21 are obtained by plotting the estimated coefficients from the logit model. The hypothetical data is generated from various brand positions, and the learned vectors are plotted to show how the model is picking up on different dimensions of the text embedding.

B Connection to recommender systems cold-start

The problem of estimating preferences for new brands has some similarities with the cold-start problem from recommender systems. Depending on the background of the reader, the cold-start problem may provide more relevant intuition for conceptualizing this paper.

The problem of recommending movies to users based on their past ratings was made

famous in the Netflix Prize (Bennett and Lanning (2007)). Given partially observed data of users and movie ratings as shown in Table 8, high-performing methods were developed for predicting how a user would rate an existing movie based on their past ratings and ratings from similar users (Resnick et al. (1994), Su and Khoshgoftaar (2009)). But recommending new movies to existing users, existing movies to new users, and new movies to new users remains difficult. As in products with strong brand effects, easily measured attributes (e.g. genre, length, budget) do not adequately represent people's preferences over movies, which makes these "cold-start" problems hard compared to the case where data can "warm-start" preference estimation. Recently, however, there has been significant progress in incorporating unstructured text data to solve the cold-start problem (Li et al. (2023), He et al. (2023)). This progress motivated the present work.

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	⋯
User 1		👍		👍	👎	⋯
User 2	👍		👍	👎		⋯
User 3	👍		?	👎	👍	⋯
User 4	👍		👍		👍	⋯
User 5		👍	👎	👎		⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table 8: The movie recommendation problem in the Netflix Prize provides helpful intuition for the brand choice setting. Given partially observed data on users and movie ratings, the goal is to infer missing ratings, e.g. the highlighted entry marked in red. Predicting ratings for new movies or new users, however, is much more difficult.

The new product introduction problem can be viewed as a type of cold-start problem but with two key differences: (1) demand-side substitution and (2) supply-side competition. If we observed sales data on brands aggregated across different markets, as shown in Table 9, introducing a new product will cause consumers to substitute away from existing products. This introduces a dependence within rows of Table 9 that was not present in the movie recommendation problem. Furthermore, firms would adjust prices to a new equilibrium after entry, creating additional dependence between a new product and existing

	Brand 1	Brand 2	$\cdots$	Brand J
Market 1	10	12	$\cdots$	9
Market 2	3	16	$\cdots$	7
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Market T	8	3	$\cdots$	12

Table 9: Hypothetical of aggregate sales across markets by competing brands within a product category. In contrast with movie ratings, entries within a row are dependent due to substitution by consumers and strategic behavior by firms (e.g. pricing). These effects must be included when modeling the entry of new brands.

products. Note that I use "brand" and "product" interchangeably following convention in the brand choice literature, though this assumption can be relaxed.

Fortunately, existing structural methods provide guidance for modeling these economic effects (Berry (1994), Berry et al. (1995)). Once demand primitives (i.e. consumer preferences) are estimated, prices are obtained by assuming firms behave optimally, and quantities are obtained by entering these prices into the demand model. To predict sales after a new product is introduced, it thus suffices to predict the demand-side primitives for the new product. Then we proceed with our modification of standard demand models to include text data.

C Uncertainty Quantification

I quantify the uncertainty of my predictions using conformal inference. The simplest implementation of conformal inference only relies on an exchangeability assumption on the residuals to derive valid prediction intervals (Lei et al. (2018)). Some issues arise with this method due to the test set data being different from the training set data, so I explore whether Bayesian approaches to uncertainty quantification can do better. I implement several methods, and the only one that improves the calibration of uncertainty is training an ensemble of predictors and using the across-model standard deviation. As discussed in Lakshminarayanan et al. (2017), this procedure can be viewed as an approximation

to the posterior predictive distribution. Two other methods I implement are repeatedly sampling from the fine-tuned LLM, as it should define some posterior distribution, and introducing randomness by randomly dropping out a fraction of the nodes in the LLM, which was shown to approximate sampling from a Bayesian posterior in Gal and Ghahramani (2016). These latter two methods do not help. I evaluate the performance of all of these methods by the width and coverage of their respective prediction intervals.

Intuitively, we want our methods to capture the notion that uncertainty should be higher when the input text is "more different" from the training data. In the literature, this is sometimes called "epistemic" uncertainty to contrast from "aleatoric" uncertainty arising due to noise. I expect that epistemic uncertainty will not be an issue for validation set uncertainty but will pose a challenge for test set uncertainty. The input text in the test set is never seen during training, which is an extreme form of covariate shift. This violates the underlying assumptions that guarantee coverage of methods like conformal inference, and I measure the severity of these violations empirically.

Suppose we have generated a prediction $\hat{\delta}_{new}$ for new text input x_{new}. To use this prediction for decision-making, some measure of the uncertainty of the prediction is helpful. That is, we want a prediction interval $\hat{C}_\alpha$ such that $Pr(\hat{\delta}_{new} \in \hat{C}_\alpha) \geq 1 - \alpha$ for any α.

Split conformal inference (Lei et al. (2018)) constructs $\hat{C}_\alpha$ from the distribution of residuals. It requires splitting the available data into training and calibration sets, and training the predictive model on the training set. Then, for each data point in the calibration set (x_{jt}, δ_{jt}), the absolute value of the residual from the predictive model is computed:

$$R_{jt} := |\hat{\delta}(x_{jt}) - \delta_{jt}|$$

Let R_{new} be the absolute value of the residual for the new input x_{new}. R_{new} is unknown, but under the assumption that the joint distribution of $(R_{jt}, \ldots, R_{new})$ is exchangeable, the rank of R_{new} will be uniformly distributed among the calibration set residuals. Or in math,

if there are N points in the calibration set,

$$Pr(R_{new} \leq \lceil (1-\alpha)(N+1) \rceil \text{ largest } R_{jt}) \geq 1-\alpha$$
$$\iff Pr\left(R_{new} \leq \frac{\lceil (1-\alpha)(N+1) \rceil}{N} \text{ quantile of } R_{jt}\right) \geq 1-\alpha$$

This is the $(1-\alpha)^{th}$ quantile of R_{jt} with a finite-sample correction. Denoting the value of this quantile by $\hat{q}_{1-\alpha}$, our prediction interval for $\hat{\delta}_{new}$ is $[\hat{\delta}_{new} - \hat{q}_{1-\alpha}, \hat{\delta}_{new} + \hat{q}_{1-\alpha}]$.

I evaluate these prediction intervals in terms of their width and coverage on both the validation and test set. The calibration set is obtained by randomly sampling half of the original validation set, with the remaining half used for validation. I present results for the width and coverage that are averaged over 30 draws of the calibration set. As shown in Figure 22, higher confidence requires wider prediction intervals. Making a prediction with 95% confidence requires a margin of ± 0.8 (for reference, the average magnitude of δ_{jt} is 3.66). Models that attain smaller residuals (i.e. they predict better) will also achieve tighter prediction intervals. The untuned LLM has the widest intervals, as it has the least accurate predictions, and the other 3 models have narrower intervals.

In Figure 23, I plot the coverage, which is the proportion of actual δ_{jt} values that are contained in the prediction interval. In the validation set, this probability equals $1-\alpha$ for all models, indicating that the exchangeability assumption on the residuals of the validation set is reasonable. In practical terms, this means that the uncertainty estimates for predictions involving existing brands can be trusted for decision-making.

In the test set, the untuned LLM has reasonable coverage, but its point predictions are too inaccurate to be used for downstream decisions. The other models have coverage less than $1-\alpha$, i.e. they are *overconfident* in their predictions. This arises due to a severe case of distribution shift where the test set is very different from the training set, which violates the exchangeability assumption justifying the validity of conformal inference. This under-coverage issue motivates a Bayesian approach to uncertainty quantification that could bet-

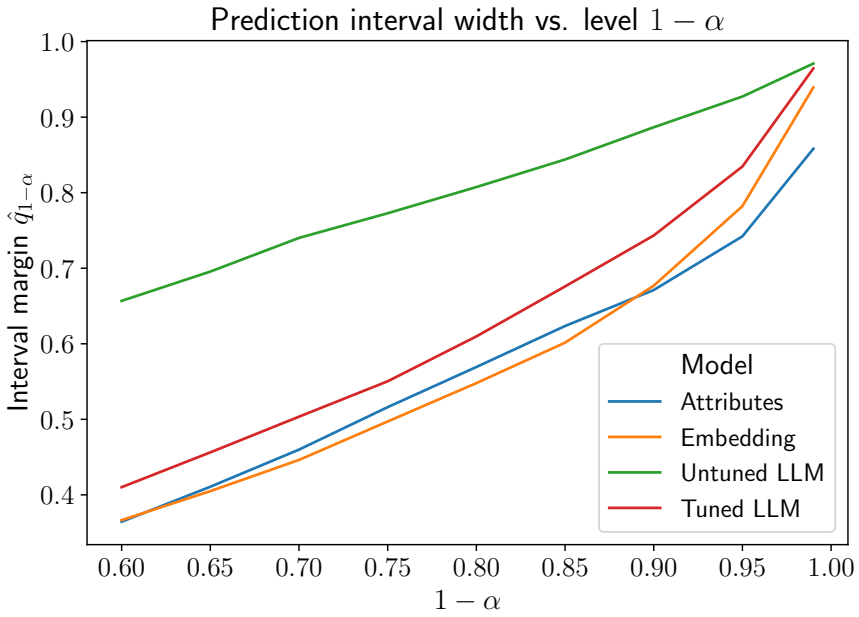


Figure 22: The width of the prediction interval from conformal inference as a function of the level $1-\alpha$. Higher confidence requires wider intervals. Making a prediction with 95% confidence requires a margin of about ± 0.8. For reference, the average magnitude of δ_{jt} is 3.66. The untuned LLM has the widest intervals because its predictions are least accurate.

ter capture the epistemic uncertainty from extrapolating outside of the training data.

I now investigate whether alternative methods of quantifying uncertainty can improve test set coverage. Suppose that our prediction model returned both a point prediction $\hat{\delta}(x)$ and an uncertainty prediction $\hat{\sigma}(x)$. As before, I split the available data into training and calibration, but this time I compute the normalized residual for each data point in the calibration set (x_{jt}, δ_{jt}):

$$\tilde{R}_{jt} := \frac{|\hat{\delta}(x_{jt}) - \delta_{jt}|}{\hat{\sigma}(x_{jt})}$$

Letting $\hat{q}_{1-\alpha}$ be the $\dfrac{\lceil (1-\alpha)(N+1) \rceil}{N}$ quantile of $\tilde{R}_{jt}$, the prediction interval for input x is

$$\hat{C}_\alpha = [\hat{\delta}(x) - \hat{q}_{1-\alpha}\hat{\sigma}(x),\ \hat{\delta}(x) + \hat{q}_{1-\alpha}\hat{\sigma}(x)]$$

In words, the uncertainty prediction $\hat{\sigma}(x)$ is *scaled* by $\hat{q}_{1-\alpha}$ to achieve the desired level. If $\hat{\sigma}(x)$ is larger for values of x outside the support of the training distribution, coverage in

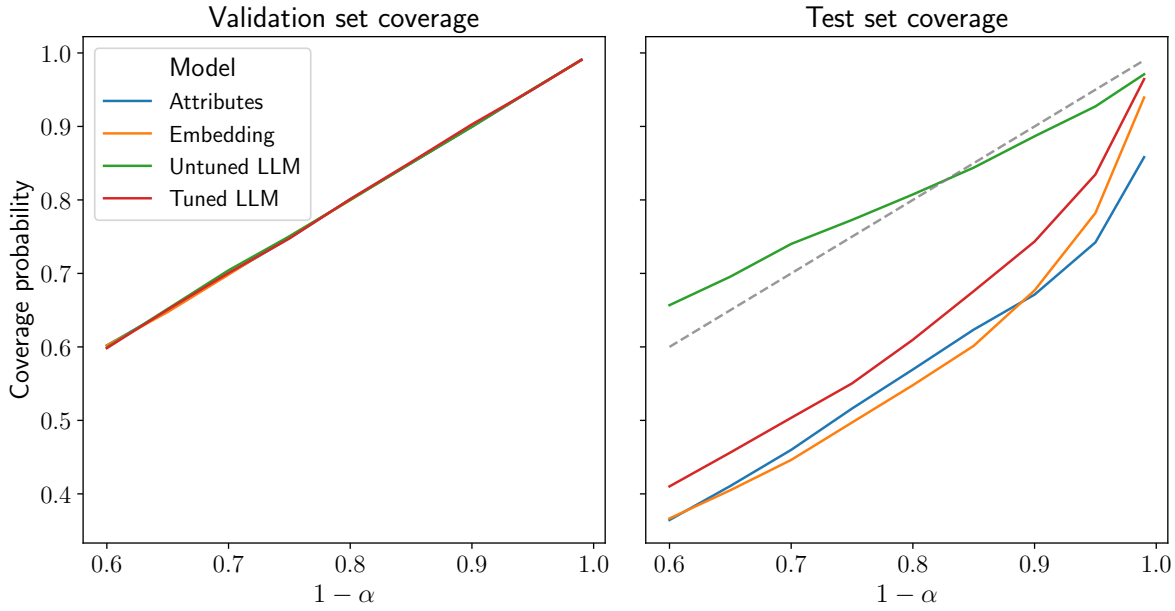


Figure 23: Coverage of the conformal inference prediction intervals in the validation and test set. The intervals have correct coverage in the validation set, but under-cover in the test set. The test set is very different from the training data, which violates the exchangeability assumption in conformal inference.

the test set should improve.

I obtain uncertainty estimates $\hat{\sigma}(x)$ by training an ensemble of base models and computing the standard deviation across their predictions. The point prediction is the average of their predictions, as is standard practice. Measuring the variation across base models was shown to effectively capture uncertainty in Lakshminarayanan et al. (2017), and the intuition is that each base model approximates a draw from the posterior predictive distribution. To evaluate whether this method improves coverage, I train a random forest, which is a simple ensemble model, on the activations of the last layer of an LLM. I compare two methods for computing prediction intervals: (1) split conformal inference on the prediction residuals, and (2) scaling the across-tree standard deviation by quantiles of the normalized residual. Since the point prediction is the same in both methods, any difference in coverage is due to the way uncertainty is quantified. For the LLM, I use the 2B parameter Gemma 2 model from DeepMind (Gemma Team (2024)). Its absolute perfor-

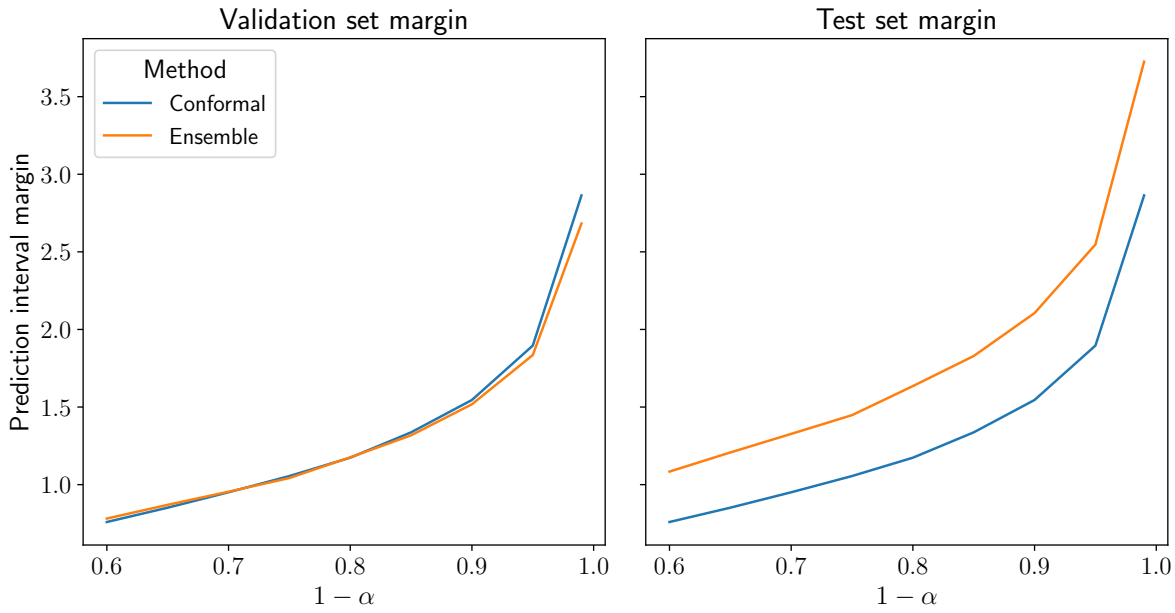


Figure 24: Validation and test set prediction margin from two methods: taking quantiles of residuals ("conformal"), and scaling the across-model standard deviation from an ensemble ("ensemble"). Ensemble correctly gives larger intervals in the test set.

mance is worse due to its small size, but it is sufficient for evaluating relative performance.

In Figure 24, both methods of computing prediction uncertainty give the same interval size in the validation set, but the ensemble method gives larger intervals in the test set. This is desirable because the test set contains data points that were unseen during training, so the model should be less sure. Consequently, the ensemble attains better test set coverage, as shown in Figure 25. While the coverage is still less than $1-\alpha$, the magnitude of the under-coverage is halved, which is significant progress. Finally, Figure 26 plots the empirical CDFs of the $\hat{\sigma}(x)$ values from the ensemble method in the validation and test sets. The test set distribution first-order stochastically dominates the validation set distribution – it is larger in every quantile. This is what we should intuitively expect of the prediction uncertainties, but other methods I tried (repeated sampling from the fine-tuned LLM and Monte Carlo dropout) failed to demonstrate this property.

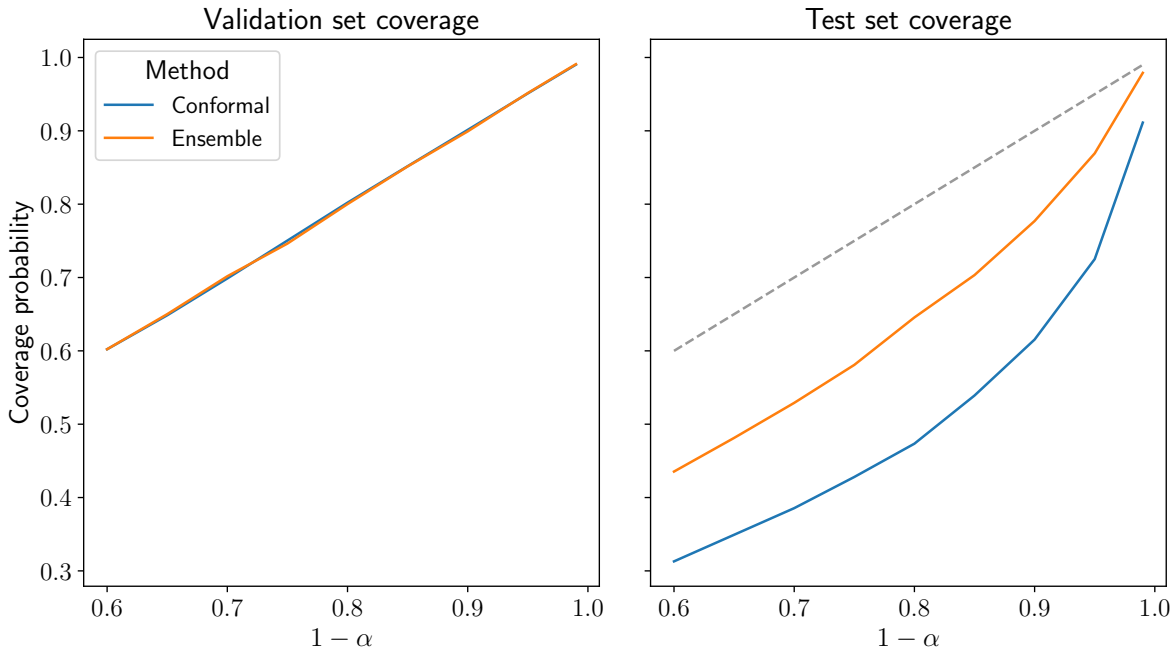


Figure 25: Coverage of the prediction intervals in the validation and test set from two methods applied to the same model. Ensemble-based method improves coverage in the test set.

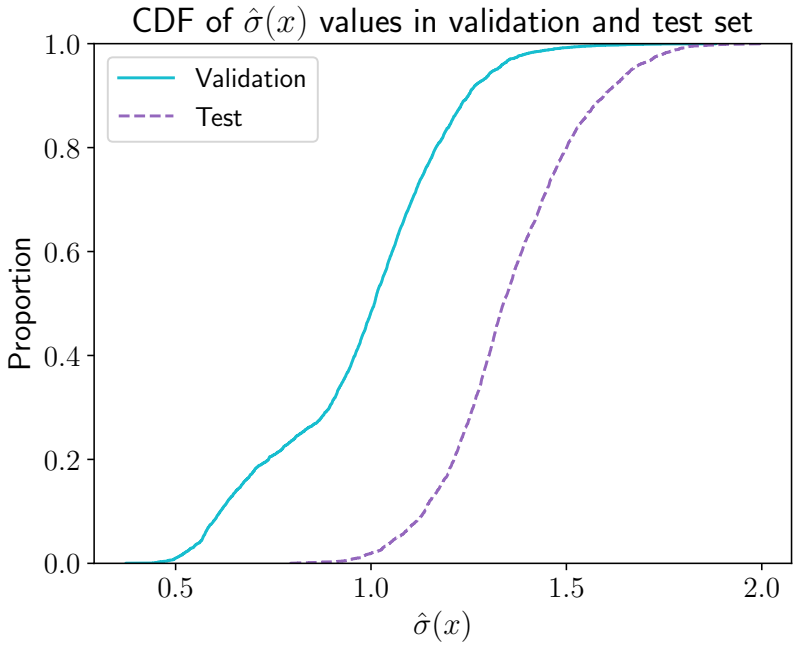


Figure 26: Distribution of validation and test uncertainties obtained from the ensemble. Every quantile is bigger in the test set, reflecting the increased uncertainty from making predictions on inputs that were not seen during training.