Even Faster Ideal Point Estimation: Using Variational Autoencoders to Estimate Item-Response Theory Models

Mason Reece*
Department of Political Science
Massachusetts Institute of Technology

July 12, 2026

Abstract

Ideal point estimation is foundational to several subfields of political science. Considerable effort has been spent to create computationally efficient methods to estimate ideal points, but these methods can still struggle on big data or complex models. I introduce variational autoencoders (VAEs), a type of machine learning model, to political science as a new method for estimating ideal points. Structuring a VAE in a particular way connects it theoretically to the classical item-response theory framework. VAEs reliably estimate ideal points and item parameters orders of magnitude faster than standard methods. They can also be easily extended to handle missing data, hierarchical structures, multiple dimensions, and dynamic models. I also develop post-estimation techniques researchers can use to "correct" VAE estimates using a subset of parameters estimated using traditional Bayesian methods. I conclude by demonstrating an empirical application using ideal point estimates from 600 million votes cast by 34.8 million voters in the 2020 U.S. election.

Word count: 6,682

1 Introduction

Ideal point estimation is foundational to several subfields of political science. Regularly, researchers have discrete data (roll call votes, social media networks, positions taken on bills, etc.) and are interested in mapping those data to a continuous, latent, quantity of interest (ideology, democratic norms, etc.). For example, scholars studying ideological positions of actors began with the spatial voting model developed in Poole and Rosenthal (1985) and have since expanded to a large number of substantive applications using a wide variety of data (e.g., Martin and Quinn, 2002; Shor and McCarty, 2011; Barberá, 2015; Bond and Messing, 2015; Lauderdale and Herzog, 2016a; Burnham, 2024; Cowburn and Sältzer, 2025; Lewis et al., 2025). The size and complexity of this data ranges from analyses of a few thousand votes in the United Nations (Bailey et al., 2017) to millions of social media followers (Barberá, 2015). The prototypical way to estimate ideal points with structured data is

*To whom correspondence should be addressed. Email: mpreece@mit.edu. Website: https://mreece13.github.io/. I thank Devin Caughey, Charles Stewart III, Andrea Campbell, Naoki Egami, Shuning Ge, Yuyang Shi, Gabrielle Péloquin-Skulski and participants in MIT's Computational Social Science Workshop for helpful comments. I acknowledge the MIT Office of Research Computing and Data for providing high performance computing resources that have contributed to the research results in this paper and to the MIT Political Methodology Lab for partially funding this research.

through the use of item-response theory (IRT) (Jackman, 2009).[1] Considerable effort has been spent to create computationally efficient methods to estimate ideal points, but these methods can still struggle on modern big data or complex models. Researchers have developed methods to estimate IRT models faster, including expectation-maximization (EM) algorithms (Chalmers, 2012; Imai et al., 2016; Peress, 2022) and variational inference (VI) methods (Grimmer, 2011; Imai et al., 2016). These methods are faster than Markov Chain Monte Carlo (MCMC) and Maximum Likelihood Estimation (MLE) techniques and represent a welcome improvement to the original methods, but they are still too slow for large data or complex models (Wu et al., 2020). For example, information on social media users (Barberá, 2015; Tai et al., 2025), campaign contributions (Bonica, 2024), or digital trace datasets (Guess et al., 2020; Stier et al., 2020) are essential to modern political science research and are still too large to estimate IRT models on.

I introduce variational autoencoders (VAEs) to political science as a new method for estimating ideal points. Although originally developed for vision and natural language processing (Kingma and Welling, 2019), VAEs are a general-purpose machine learning model that can be structured to work for many types of problems. Recent psychometric work has shown that VAEs can be structured in a particular way such that they are theoretically tied to the IRT framework and perform well in empirical applications (Converse et al., 2021; Curi et al., 2019; Ma et al., 2024; Cho et al., 2021). Mechanically, instead of fitting a separate function for each ideal point, VAEs fit a single, flexible, function on the data, which is then used to predict the ideal point for each voter. Because the size and complexity of the flexible function is independent of the number of individuals, VAEs can be orders of magnitude faster than traditional methods. The size of the speedup can vary based on the problem, but in the empirical example I explore below, estimates from a VAE correlate with the estimates of ideal points from traditional MCMC algorithms at 0.95 and the algorithm runs 5,000 times faster. The parameters of the flexible function can also be interpreted as the item parameters in a standard ideal point model. These parameters are automatically estimated as part of the VAE, and I show examples of this use case in the application section.

This paper unifies prior work on VAEs and extends it to handle common ideal point estimation tasks in political science. I show how VAEs can be structured to handle item- or individual-level covariates, missing data, hierarchical structures, multiple dimensions, and dynamic models. Although these topics have each been the subject of independent research in past work on ideal point estimation, I show how they can all be efficiently implemented under the same VAE framework.

Despite efforts to theoretically align VAEs with traditional MCMC algorithms for IRT models, VAEs, or any other approximation method, may be non-randomly biased. Since the method is both unsupervised and it is not feasible to estimate the full model using MCMC, I propose a correction on downstream estimates based on the work of Egami et al. (2024). Researchers would estimate a subset of ideal points using MCMC on a strategically chosen sample of data, then use a predictive model between these "gold standard" estimates and the VAE estimates to debias downstream uncertainty estimates. Although here I only show this method for VAEs, the method could also be generally applied to other approximation methods, such as variational inference (Grimmer, 2011).

I conclude by demonstrating an empirical application of the VAE model with nearly 600 million votes cast in the 2020 U.S. election. Using the model, I'm able to estimate ideal points for 34.8 million voters. I demonstrate how these estimates are fundamentally different from other voter-level ideology estimates and use them to reassess representation in state legislative elections from the 2020 election (Rogers, 2017).

[1] Researchers also regularly estimate ideal points using unstructured data (Abi-Hassan et al., 2023; Herrmann and Döring, 2023; Lauderdale and Herzog, 2016b; Peterson and Spirling, 2018; Laver et al., 2003; Däubler and Benoit, 2017; Slapin and Proksch, 2008). This is a large, and growing, field in political science, especially with the recent adoption of large-language models. However, estimation of ideal points with structured data is still popular, both because the data used are still relevant and because the methods used are parametric in nature and easier to verify and interpret than many of the black-box methods used for unstructured data.

2 Item-Response Theory

First, a brief overview of the traditional methods to estimate ideal points following Item-Response Theory in the context of political science. IRT models in political science are closely connected to the spatial voting model, where a voter is assumed to choose an option if the utility a voter derives from that option is greater than any other option (Enelow and Hinich, 1984). As a running example, related to the empirical application in Section 4, consider data on the candidate a given voter voted for in each of the contests on their ballot. Jackman (2009) shows that beginning with the spatial voting model plus the assumption that any stochastic error about this utility is normally distributed results in the following model for the probability that a candidate is selected by a voter:

$$\pi_{jkc} = Pr(y_{jk} = c \mid \alpha_j, \{\gamma_{kc'}, \beta_{kc'}\}_{c'=1}^{C_k}) = \frac{\exp(\eta_{jkc})}{\sum_{c'=1}^{C_k} \exp(\eta_{jkc'})}, \quad \text{where } \eta_{jkc} = \alpha_j \gamma_{kc} - \beta_{kc}, \tag{1}$$

where j indexes voters, k indexes the contest, c indexes the candidate in that contest, C_k is the number of candidates in contest k, and η_{jkc} is the *linear predictor* for candidate c.[2] This model is the canonical two-parameter IRT model. The function mapping the linear predictors to a probability is the softmax, a multivariate generalization of the inverse logit; this choice reflects the reality that voters often have more than two candidates to choose among on the ballot. Note that the probability of choosing candidate c depends on the full set of parameters $\{\gamma_{kc'}, \beta_{kc'}\}_{c'=1}^{C_k}$ for *every* candidate in contest k, not just those of candidate c. There are three categories of parameters of interest. First, $\alpha_j \in \mathbb{R}$ is the unobserved ideal point of voter j. Second, γ_{kc} is an unknown *item discrimination* or slope parameter for each candidate who could be chosen in the contest. γ_{kc} describes the degree to which a vote for a candidate in the contest informs the model's placement of a voter's ideal point. Third, β_{kc} is an *item difficulty* parameter which governs the baseline log-odds of choosing candidate c irrespective of ideology. A higher β_{kc} corresponds to a *lower* baseline probability of selection, since it enters the linear predictor with a negative sign. Identifying the model requires resolving two distinct issues. First, within each contest the softmax is invariant to adding a constant to every η_{jkc}, so one candidate must serve as a reference category, with its γ_{kc} and β_{kc} set to 0; in the categorical case, where it may be hard *a priori* to know the direction of effects, this reference category is usually chosen randomly. Second, and separately, the latent space itself has rotational, reflection, and scale indeterminacy, which is resolved with standard constraints on the decoder, anchor items, and/or post-processing (Rivers, 2003).

Combining the choice probabilities into a full likelihood yields the following model:

$$L(\Omega, \alpha_j | \mathbf{y}_j) = \prod_{k=1}^{K} \pi_{jky_{jk}} = \prod_{k=1}^{K} F(\alpha_j \gamma_{ky_{jk}} - \beta_{ky_{jk}}), \tag{2}$$

where Ω includes both the discrimination and difficulty parameters and the notation $[\cdot]_{y_{jk}}$ is used to indicate which candidate voter j actually chose in contest k. Writing the likelihood as a product over contests k assumes that choices are conditionally independent across contests given the voter's ideal point $\boldsymbol{\alpha}_j$.

There are a number of algorithms to estimate all of the parameters of interest, including Hamiltonian Monte Carlo (HMC) as implemented in Stan, expectation maximization (EM) algorithms for certain types of data (Chalmers, 2012; Imai et al., 2016; Peress, 2022), and variational inference (VI) methods (Grimmer, 2011; Imai et al., 2016). These methods differ in how they confront the intractable marginal likelihood

[2] Although I don't show it here for clarity, the model can also be multidimensional. The additional dimension is reflected in the ideal point α_{jd} and in the discrimination parameter, γ_{kcd} where d indexes the latent dimension. The difficulty parameter, β_{kc}, is the same for all dimensions.

$p(\boldsymbol{y}; \boldsymbol{\Omega}) = \int p(\boldsymbol{y}|\boldsymbol{\alpha}; \boldsymbol{\Omega}) p(\boldsymbol{\alpha}) d\boldsymbol{\alpha}$ (note that $p(\boldsymbol{y}|\boldsymbol{\alpha})$ is the *conditional* likelihood, not the marginal). EM maximizes the marginal likelihood $p(\boldsymbol{y}; \boldsymbol{\Omega})$ by integrating out $\boldsymbol{\alpha}$; HMC instead samples directly from the joint posterior; and VI learns an approximating distribution $q(\boldsymbol{\alpha})$ for the posterior by maximizing the ELBO (see Appendix Appendix A for a formal comparison). VI has theoretical guarantees because it seeks to maximize the evidence lower bound (ELBO), formally defined as:

$$\text{ELBO} = \mathbb{E}_{q(\boldsymbol{\alpha})}[\log p(\boldsymbol{Y}|\boldsymbol{\alpha}; \boldsymbol{\Omega})] - \text{KL}[q(\boldsymbol{\alpha}) \,||\, p(\boldsymbol{\alpha})], \tag{3}$$

where $\text{KL}(\cdot)$ is the Kullback–Leibler divergence between two distributions. I apply Bayes Rule to Eq. 3 and rearrange:

$$\log p(\boldsymbol{Y}) = \text{ELBO} + \text{KL}[q(\boldsymbol{\alpha}) \,||\, p(\boldsymbol{\alpha}|\boldsymbol{Y})]. \tag{4}$$

(see Appendix Appendix A for the derivation). Note that the KL term in Eq. 3 is taken against the *prior* $p(\boldsymbol{\alpha})$, whereas the KL term in Eq. 4 is taken against the *posterior* $p(\boldsymbol{\alpha}|\boldsymbol{Y})$. Therefore, as the function $q(\boldsymbol{\alpha})$ more closely resembles $p(\boldsymbol{\alpha}|\boldsymbol{Y})$, the ELBO more tightly bounds the marginal likelihood. By selecting a simpler $q(\boldsymbol{\alpha})$, VI makes estimation a more tractable problem. The ELBO is maximized over the variational parameters ϕ, $q^* = \text{argmax}_{q \in Q}$ ELBO, and the ideal point estimate is then the variational mean $\hat{\alpha}_j = \mathbb{E}_{q^*}[\alpha_j]$. Although I only reference the ideal point $\boldsymbol{\alpha}$ here, researchers can also fit approximation functions for each of the other parameters in the model (Grimmer, 2011).

Unfortunately, HMC, EM, and VI are all still insufficient for large data sets. To see why, consider the data set of cast vote records released in Kuriwaki et al. (2024). Cast vote records are digital representations of a ballot cast by a voter and they contain the candidate each voter voted for in each contest on the ballot. They match perfectly with the setup at the start of this section. The data set includes the ballots from 34.8 million voters and 7,164 contests. Estimating ideal points using any of the three methods outlined above would involve estimating a posterior, or an approximation of a posterior, for every single voter. Furthermore, for each additional contest, I need to also calculate a discrimination and a difficulty parameter. As the number of parameters increases, the estimation of each individual parameter becomes more challenging since I am estimating a joint distribution. Finally, if researchers are interested in estimating additional latent dimensions, the problem increases by a factor of the number of additional dimensions. Although technically possible, the computation of this model can be time and resource intensive. For medium sized data, some researchers may be happy to simply wait the computational time required. But, for larger data, or in cases where researchers may want to iterate, the time cost can be prohibitive. This problem has encouraged some researchers to resort to simpler estimation methods (e.g., Lai et al., 2024). In the next section I introduce a new, faster, method – variational autoencoders.

3 Variational Autoencoders

Variational autoencoders (VAEs) are closely connected to variational inference. The difference is that instead of choosing a distribution $q(\boldsymbol{\alpha})$ to approximate each posterior, I instead build a single flexible inferential model that predicts the posteriors for each ideal point. The parameters of this model are shared by all observations in the data, making the size and complexity of the model or the number of parameters to estimate simply a function of researcher choice instead of scaling with the number of observations. By carefully constructing the model such that it is both interpretable and sufficiently flexible, I can recover all parameters of interest from the standard IRT model.

In general, VAEs are an unsupervised machine learning method. Intuitively, an autoencoder *encodes* the input (the choices of each voter in each contest) to a latent dimension(s) using a learned neural network, then *decodes* this latent representation to reconstruct the original input using a second, connected, learned neural network. Autoencoders have been in use in the machine learning literature for over a decade. Originally developed for image recognition, their use has since expanded into other areas (Kingma and Welling, 2019). I make the autoencoder "variational" by using it to approximate the true posterior.

Formally, I approximate the posterior distribution of the latent variables with a multivariate normal *variational distribution* (the approximate posterior):

$$q_\phi(\boldsymbol{\alpha}_j \mid \boldsymbol{y}_j) = \mathcal{N}(\boldsymbol{\mu}_j, \text{diag}(\boldsymbol{\sigma}_j^2)), \text{ with } (\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j) = \text{NeuralNet}_\phi(\boldsymbol{y}_j), \tag{5}$$

where $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_j$ characterize the mean and standard deviation of the multivariate normal (so $\text{diag}(\boldsymbol{\sigma}_j^2)$ places the variances on the diagonal), and ϕ contains all the parameters of the neural network model. Together, these constitute the entire "encoder" part of the VAE. To optimize the new parameters I now maximize a slight variation of the ELBO shown in Eq. 3:

$$\text{ELBO} = \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\boldsymbol{Y})}[\log p(\boldsymbol{Y}|\boldsymbol{\alpha}; \boldsymbol{\Omega})] - \text{KL}[q_\phi(\boldsymbol{\alpha}|\boldsymbol{Y}) \mid\mid p(\boldsymbol{\alpha})], \tag{6}$$

where the approximate posterior $q_\phi(\boldsymbol{\alpha}|\boldsymbol{Y})$ is conditioned on the data $\boldsymbol{Y}$ through the encoder (the defining feature of *amortized* inference), and where optimization is performed jointly over the network parameters ϕ and the decoder parameters $\boldsymbol{\Omega}$ rather than directly over voter-specific posteriors.

A VAE model is trained by minimizing the loss between the original data and the learned data construction, where the parameters of both the decoder and encoder networks are adjusted using standard stochastic gradient descent methods and backpropagation. I reparameterize the model to ensure backpropagation can properly occur because the gradient cannot be passed through an expectation taken with respect to a ϕ-dependent distribution; score-function (REINFORCE) estimators exist but have prohibitively high variance (see Appendix Appendix A) (Wu et al., 2020; Urban and Bauer, 2021). To see this, note that

$$\begin{aligned} \nabla_\phi \text{ELBO} &= \nabla_\phi \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\boldsymbol{Y})}[\log p(\boldsymbol{Y}|\boldsymbol{\alpha}; \boldsymbol{\Omega}) + \log p(\boldsymbol{\alpha}) - \log q_\phi(\boldsymbol{\alpha}|\boldsymbol{Y})] \\ &\neq \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\boldsymbol{Y})}[\nabla_\phi \log p(\boldsymbol{Y}|\boldsymbol{\alpha}; \boldsymbol{\Omega}) + \nabla_\phi \log p(\boldsymbol{\alpha}) - \nabla_\phi \log q_\phi(\boldsymbol{\alpha}|\boldsymbol{Y})], \end{aligned} \tag{7}$$

because the expectations are taken with respect to $q_\phi(\boldsymbol{\alpha}|\boldsymbol{Y})$, which is a function of ϕ. This problem is resolved by reparameterizing $\boldsymbol{\alpha}$ as

$$\begin{aligned} \boldsymbol{\varepsilon} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D) \\ \boldsymbol{\alpha} &= \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}, \end{aligned} \tag{8}$$

where $\boldsymbol{\varepsilon}$ is a $D \times 1$ sample from a standard multivariate normal distribution, D is the number of latent dimensions, and $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are outputs from the NeuralNet above. This isolates the randomness into $\boldsymbol{\varepsilon}$, so that $\boldsymbol{\alpha}$ is a deterministic function of the NeuralNet parameters ϕ given $\boldsymbol{\varepsilon}$. Therefore, I can now calculate the second line in Eq. 7 where the expectation is now taken with respect to $p(\boldsymbol{\varepsilon}) = \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$. These models can easily be implemented in PyTorch, Keras, or any other standard machine learning library.

The objective above can be tightened by drawing multiple importance-weighted samples. With S samples $\boldsymbol{\alpha}^{(s)} \sim q_\phi(\boldsymbol{\alpha}|\boldsymbol{y})$, the importance-weighted ELBO (IW-ELBO) is

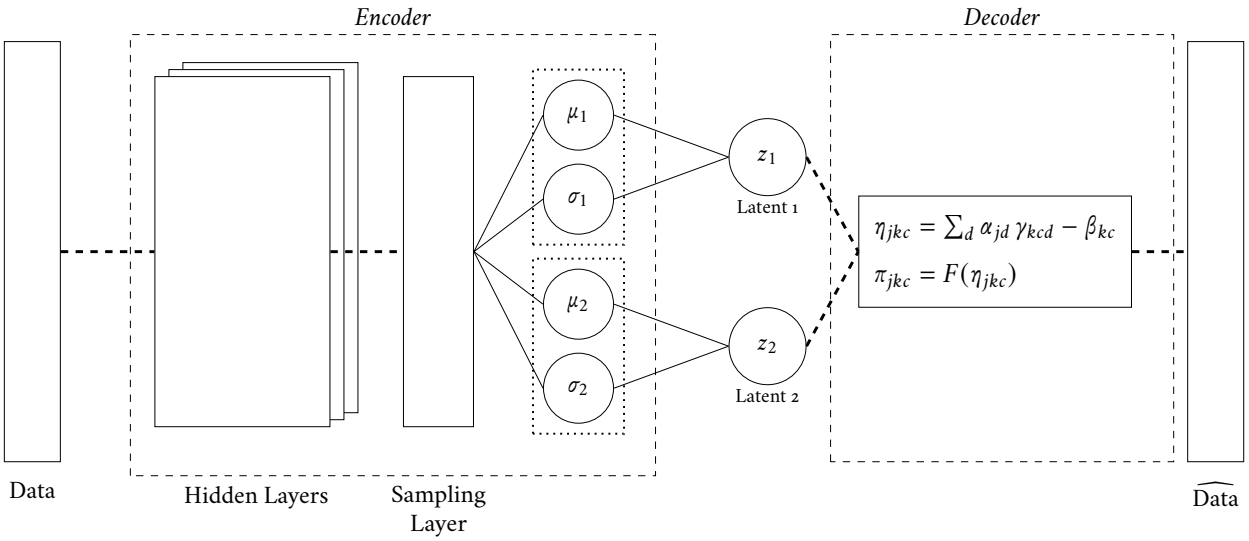

Figure 1: Architecture of a 2-Dimension VAE. Data enter the encoder, which maps ballots through hidden layers and an output layer to per-dimension distribution parameters $(\mu_k, \log \sigma_k^2)$ (grouped by latent dimension in dotted boxes). The reparameterization layer then yields latent samples $z_k = \mu_k + \sigma_k \varepsilon_k$, $\varepsilon_k \sim \mathcal{N}(0,1)$, which serve as the voter ideal points α_{jd}. The single-layer IRT decoder forms the linear predictor $\eta_{jkc} = \sum_d \alpha_{jd}\,\gamma_{kcd} - \beta_{kc}$ and maps it through the softmax $F(\cdot)$ to the choice probability π_{jkc} (as in Eq. 1), reconstructing $\widehat{\text{Data}}$. Heavy dashed lines denote densely connected layers; thin solid lines denote individual parameter connections.

$$\mathcal{L}_S = \mathbb{E}\left[\log \frac{1}{S}\sum_{s=1}^{S} w_s\right], \quad \text{where } w_s = \frac{p(\boldsymbol{y}|\boldsymbol{\alpha}^{(s)};\boldsymbol{\Omega})\; p(\boldsymbol{\alpha}^{(s)})}{q_\phi(\boldsymbol{\alpha}^{(s)}|\boldsymbol{y})}. \tag{9}$$

The IW-ELBO is a lower bound on the marginal log-likelihood, $\mathcal{L}_S \leq \log p(\boldsymbol{y})$, and the bound tightens monotonically as $S \to \infty$.[3] Setting $S = 1$ recovers the standard ELBO.

Figure 1 visually represents a basic VAE with two latent dimensions. Beginning from the left side of the diagram, data are fed as inputs to the encoder model, which is composed of some number of hidden layers before passing through an output layer. The output layer produces the parameters μ and $\log \sigma^2$ when a normal distribution is imposed; the reparameterization layer then samples $\alpha = \mu + \sigma \odot \epsilon$. Those latent dimensions are then passed through the decoder network to reconstruct the original data.

There are a couple points to make about the connection between the VAE and IRT model. First, VAEs add an assumption that the latent dimension(s) follow some probability distribution, in many cases, a multivariate normal distribution. Notice the similarity between this assumption and the assumption in a classical spatial voting model that errors are distributed normally. This assumption is encoded as an output layer at the end of the encoding model which then outputs parameters describing the distribution of the model (in the standard normal case, parameters μ and $\log \sigma^2$). The second modification is that the decoder network has only a single layer that passes the learned latent variables through the exact same $F(\cdot)$ function as in Eq. 1. With this single-layer decoder, the linear predictor for candidate c in contest k is $z_{kc} = \sum_d \gamma_{kcd}\,\alpha_{jd} - \beta_{kc}$, after which $F(\cdot)$ maps the predictors to choice probabilities. Here the decoder *weights* γ_{kcd} multiply the latent ideal point $\boldsymbol{\alpha}_j$, and the (negated) decoder *biases* β_{kc} are the intercepts (cf. Converse et al., 2021). Notice that the decoder

[3]The IW-ELBO was introduced as the importance-weighted autoencoder objective; with $S = 1$ it recovers the standard single-sample ELBO above.

weights correspond exactly to the discrimination parameters γ, and the decoder biases correspond to the (negated) difficulty parameters β, of the IRT setup – so the weights and biases of the VAE can be interpreted in the same manner as the discrimination and difficulty parameters in the IRT model (Curi et al., 2019).

Model Architecture

Of course, a key component of all neural networks is how to construct the model architecture. Unfortunately, there are not absolute recommendations, but I offer and validate suggestions. For the hidden layers in the encoder, I recommend multiple hidden layers and to choose a size for each hidden layer as a multiple of 2 (in the models I fit below, I choose 64). The size and number of these hidden layers is a balancing act of computational time and GPU memory – more and larger layers tend to be more "flexible" models but are also more computationally intensive to fit. In machine learning it is also common to batch responses together, a process called "mini-batching." Again, larger batches are generally "better" but more computationally challenging to evaluate. Here, I use batches of size 256. Another choice is the embedding dimension, the length of the vector that each contest's embedding table uses to represent the candidate a voter chose before these vectors are pooled into the encoder; larger embeddings give the model more capacity to encode each vote but add parameters and, like larger hidden layers, raise computational cost. I set an embedding dimension of 16 for the main model. I set a standard learning rate of 0.001. I only take one importance-weighted sample, reducing Eq. 9 to just the ELBO. In Appendix C I show that the choice of hyperparameters do not substantively change the correlation between a VAE and MCMC model. The main trade-off is in computational time and GPU memory – larger models with more hidden layers or larger batches take more time to fit and require more computer memory.

3.1 Extensions

The literature on VAEs in general is rich and contains many optimizations to the basic model presented here or specific adaptations for other circumstances. Here, I overview several relevant extensions for political scientists: missing data, correlated latent dimensions, temporal dynamics, and the incorporation of contest-level and voter-level covariates. Each preserves the amortized encoder and the IRT-interpretable single-layer decoder developed above, changing only the pieces of the model that the extension requires. The extensions are also not mutually exclusive, so researchers can implement any number of them at once depending on their case.

Missing Data

Missing data are common in political science applications, but unfortunately due to its neural network structure are a challenging application for a VAE. The problem lies in the step within a neural network where gradients are computed with respect to each of the network parameters. Where inputs are missing, the network has no principled value to propagate, and including imputed values in the reconstruction loss biases the gradients. Similar to other missing data work, listwise deletion or otherwise just dropping missing data is not advised for statistical efficiency in general. In the running example of voter level ideal points based on their votes on their ballot, deletion is impossible – across the full set of voters only the presidential contest is shared by all voters. All contests that a voter could not have cast a ballot in are labeled "missing." Previous work on voter level ideal points has obviated this problem by constructing the data such that all voters have the same choice set (e.g., Lewis, 2001). Models to handle this kind of data are only rarely addressed in political science, with one notable exception constructing a different kind of multinomial logit estimator for the varying choices (Yamamoto, 2014).

There are several solutions to missing data proposed in the VAE literature. Veldkamp et al. (2025) review

the options and conduct simulation of each method. Of the methods they tested, the "conditional VAE" had the lowest error compared to the true values and was also one of the fastest methods. Conditional VAEs are intuitively simple. Instead of dropping missing data or imputing missingness, I can simply pass a mask of 1s and 0s to the VAE as an additional input (Collier et al., 2020). The mask is used in two steps – first in the initial hidden layers to zero out any inputs that are missing, and then again in the loss function to ensure that loss is only computed for values that are actually present in the data. The resulting masked loss function is derived formally in Appendix Appendix A.

Correlated Latent Dimensions

The current implementation of the VAE forces each of the latent dimensions to be independent. In cases where this might be an unreasonable assumption, researchers could instead allow for, and directly model, the correlation between latent dimensions. For example, in studies of American politics it might be reasonable to allow the first dimension, typically assumed to be ideology, to also correlate with additional dimensions – e.g., economic values or social policy. From a modeling perspective, the change is subtle but impactful. Instead of training my model to fit the latent space to the standard normal distribution, $\mathcal{N}(\mathbf{0}, \mathbf{I})$, researchers can generalize the prior to $\mathcal{N}(\mathbf{0}, \Sigma_0)$, where Σ_0 is a $D \times D$ covariance matrix with entries for each pair of latent dimensions (Converse et al., 2021). This generalization requires additional constraints to resolve the rotational identification of the latent space, and the KL term in the ELBO no longer has the simple closed form available in the diagonal case.

Temporal Dynamics

Ideal-point models are frequently used to track change over time, whether in the composition of an electorate or in the movement of the actors on a fixed scale. The obstacle is comparability: a different set of contests is held in each election, so ideal points estimated period-by-period live on arbitrary, incomparable scales. The dynamic IRT literature typically solves this by imposing a random-walk prior that ties each actor's position in period t to its position in period $t-1$ (Martin and Quinn, 2002). That approach presumes the *same* units are observed repeatedly, which cast vote records do not satisfy: ballots are anonymous, so a voter cannot be linked across elections, and each period is instead a repeated cross-section of a distinct electorate.

I therefore identify a common scale across time through *bridging* rather than a temporal transition prior, following the logic of bridging observations in dynamic IRT (Bailey et al., 2017). Candidates or offices that recur across elections – a returning incumbent, a persistent party label – serve as bridges: their item parameters are shared across every period in which they appear, anchoring the otherwise separate period-specific latent spaces to a single comparable scale. Concretely, the model keeps one decoder over the pooled candidate vocabulary of all elections. Each candidate c owns a single set of item parameters $(\gamma_{kcd}, \beta_{kc})$; a bridging candidate reuses those same parameters in each period it contests, while a period-specific candidate has parameters that are only ever used in that period.

The encoder, by contrast, is period-specific. Because each election presents a different menu of contests, I fit a separate inference network ϕ_t per period,

$$(\boldsymbol{\mu}_j^{(t)}, \boldsymbol{\sigma}_j^{(t)}) = \text{NeuralNet}_{\phi_t}(\boldsymbol{y}_j^{(t)}), \tag{10}$$

that maps the ballot voter j cast in period t to that voter's ideal point $\boldsymbol{\alpha}_j^{(t)}$. The decoder then forms the same IRT linear predictor as before, $\eta_{jkc}^{(t)} = \sum_d \alpha_{jd}^{(t)} \gamma_{kcd} - \beta_{kc}$, with the shared parameters of the bridging candidates supplying the cross-period link. The prior on each period's ideal points remains $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and the training

objective is the same importance-weighted ELBO pooled across periods; mini-batches are drawn one period at a time so that the matching encoder and the correct candidate parameters are used for each batch. Finally, because the latent scale is only identified up to reflection, I orient each period *post hoc* against a fixed partisan anchor (a Democratic U.S. House candidate) so that the scale points in the same direction in every period.

Contest-Level Covariates

Researchers often know a great deal about the *contests* themselves – the office at stake, whether it is partisan, its geographic scope – and may wish to let these observed features inform the model. I incorporate contest-level (item) covariates by augmenting the decoder while leaving the encoder, prior, and data pipeline untouched. For contest k, let $\boldsymbol{c}_k$ be a fixed vector of observed characteristics. The decoder concatenates it to the latent ideal point before forming each candidate's logit,

$$\eta_{jkc} = \sum_d \alpha_{jd}\, \gamma_{kcd} + \boldsymbol{c}_k^\top \boldsymbol{\delta}_{kc} - \beta_{kc}, \tag{11}$$

where $\boldsymbol{\delta}_{kc}$ is a learned coefficient vector on the contest features. The covariates $\boldsymbol{c}_k$ enter as fixed inputs (one row per contest) and are not themselves estimated; only their coefficients $\boldsymbol{\delta}_{kc}$ are. Because $\boldsymbol{c}_k$ is identical for every voter in the contest, the covariate term does not interact with $\boldsymbol{\alpha}_j$: it enters as a structured, voter-invariant shift to each candidate's baseline support, letting observed contest features help explain the difficulty parameters rather than estimating each one freely. This composes with the discrimination structure that governs which latent dimensions a contest loads on, which can aid interpretation and identification in the multidimensional models discussed above (Shin, 2025; Morucci et al., 2024).

Voter-Level Covariates

Symmetrically, one may wish to condition on characteristics of the *voter*. With cast vote records this is necessarily limited: ballots are anonymous, so the only available covariates are contextual – attached to a ballot's precinct or county (e.g., urbanicity, median income, or aggregate demographics) – rather than to the individual. Where such covariates are available, the natural place to introduce them is the *prior* rather than the decoder, mirroring hierarchical IRT. Let $\boldsymbol{x}_j$ collect the covariates attached to voter j. In place of the exchangeable prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$, I use a conditional prior whose mean depends on the covariates,

$$p(\boldsymbol{\alpha}_j \mid \boldsymbol{x}_j) = \mathcal{N}(\mathbf{B}\boldsymbol{x}_j, \Sigma_0), \tag{12}$$

where $\mathbf{B}$ maps covariates into the latent space (and Σ_0 may be the identity, or the full covariance of the previous extension). This is the ideal-point analogue of a regression on the latent trait: observed context shifts a voter's *expected* position, while the ballot still drives the posterior through the encoder. The only change to the objective is in the KL term, now taken against this covariate-dependent prior rather than the standard normal; the encoder, decoder, and reparameterization are unchanged. This differs from the contest-covariate case deliberately – voter attributes should shift *who* a voter is expected to be, so they enter the prior, whereas contest attributes shift *how* a contest behaves, so they enter the decoder. Unlike the two preceding extensions, voter-level conditioning is a design recommendation rather than a component of the models I fit below, and I leave its empirical exploration to future work.

3.2 Correcting Bias

Estimates of ideal points from VAEs are highly accurate, but may still be biased in non-random ways. It is impossible for researchers to know this a priori, especially given the unsupervised nature of the method. Therefore, I suggest researchers use the following procedure to debias their estimates, adhering closely to the suggestions in Egami et al. (2024). First, researchers should estimate an MCMC model on a small random subset of the data. This will be used as a "ground truth" to evaluate the VAE estimates. Selecting the random subset can be challenging, but I recommend trying to get a reasonable number of units within each "item" of interest.[4] This process may require some trial and error to balance the computational time needed for MCMC with coverage. Second, researchers should estimate a VAE model on the full data. Third, using the overlapping points from the first two steps, researchers should build a predictive model that maps the VAE estimates to the MCMC estimates. The choice of prediction model is important, but likely follows from standard recommendations for predictive models – choose a flexible model and rely on cross-fitting to avoid overfitting. Finally, using that predictive model, researchers can correct the remaining VAE estimates following the procedure in Egami et al. (2024). This process will allow researchers to use VAE estimates as if they were MCMC estimates, with the added benefit of being able to use all of the data instead of just a subset.

4 Empirical Application

I conclude with an empirical application using ballot-level data called cast vote records (CVRs) from voters in the U.S.'s 2020 general election. CVRs are anonymized records of all the choices made by each voter on their ballot. The data has been standardized into a common format across the nation to facilitate analysis and vote totals have been verified.[5] See Appendix B.1 for more details on the data and additional data cleaning I perform on the data for analysis. The resulting set of CVRs covers 17 states, 289 counties, 34.8 million voters and 7,164 unique offices. Since each voter cast ballots in several contests, there are a total of 594.3 million votes in the data.

CVRs have been used to study voter behavior in a number of contexts before (e.g., Lewis, 2001; Wand et al., 2001; Herron and Sekhon, 2003; Gerber and Lewis, 2004; Herron and Lewis, 2007; Frisina et al., 2008; Bafumi et al., 2012; Alvarez et al., 2018; Morse, 2021; Reece et al., 2024; Conevska and Mutlu, 2025; Dowling et al., 2025; Kuriwaki, 2026). Here, I use them to estimate a latent ideal point for each voter in the data. The ideal point estimates will then be used to assess the ideological composition of each district in the data. The most closely related work is Lewis (2001), where the authors estimate ideal points using the ballot measure choices of voters in Los Angeles and compares them to their presidential vote choice, as an estimate of split-ticket voting. Here, I extend that work by using the full ballot for estimation and greatly expanding the scope of the analysis.

4.1 Estimation

I then estimate two models, one using traditional MCMC on a random sample of voters from Adams County, Colorado and the second using a VAE on the full data. Adams County was also selected randomly but is a reasonably politically and demographically diverse county in Colorado. For more details on the traditional model, see Appendix B.2. For more details on the architecture of the VAE, see Appendix B.3. As a rough comparison in computation time, it took the MCMC algorithm 3 hours and 45 minutes to fit a single-dimensional model on the Adams County data (250,000 votes from 12,000 voters). Fitting a model on the full data (594.3 million votes from 34.8 million voters) takes 140 hours, a significant but manageable time.

[4] In the case of voter ideal points, each contest is an item, so I sample random voters from within each contest. In the case of legislator ideal points, researchers would sample random roll call votes from each legislators.

[5] More details on this process can be found in Kuriwaki et al. (2024). I extend Kuriwaki et al. (2024) to all local contests.

4.2 Validation

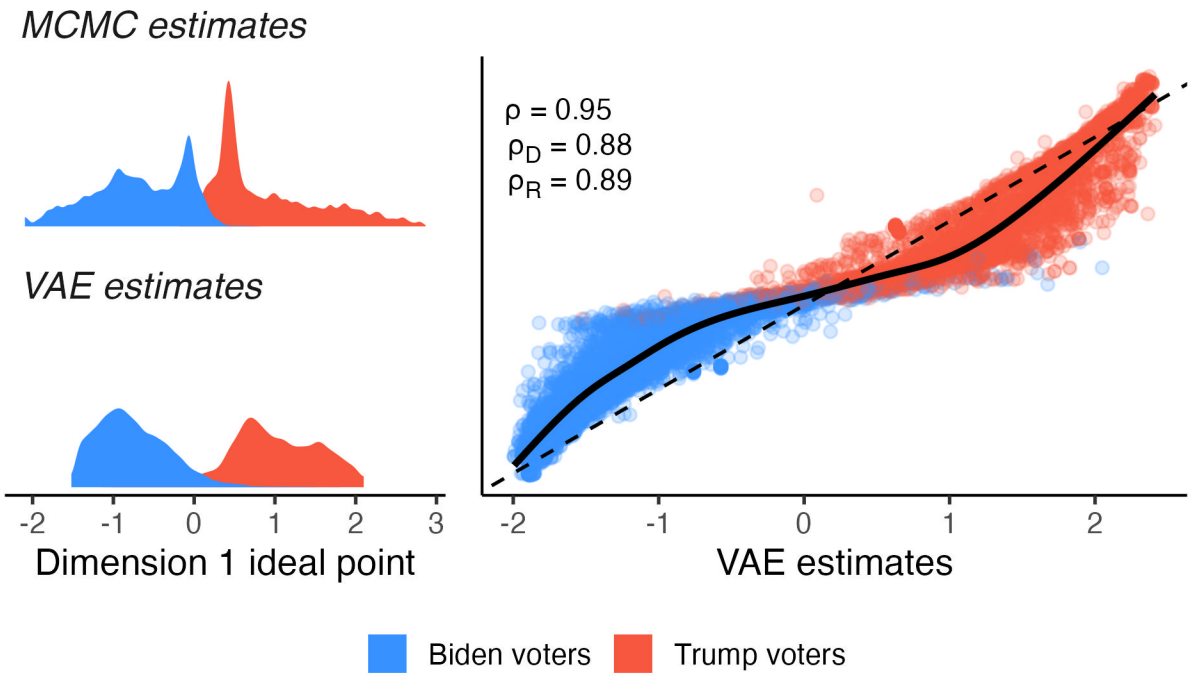

Figure 2: Comparison of MCMC and VAE Methods. The left plots display the the densities of the mean ideal point for voters who selected Joseph R Biden or Donald J Trump in the presidential contest. The right plot directly plots each matched voter's mean ideal point from the VAE algorithm and the MCMC algorithm. The solid black line is a loess line fit to the data and the dashed black line is the 45 degree line. Correlation coefficients for all the data, just for Democrats, and just for Republicans are reported in the top-left corner of the right plot.

My initial visual inspection of the VAE model is positive. In Figure 2, I plot ideal point estimates from both the MCMC and VAE model. The VAE model's ideal points have been subset down and matched to only those ideal point estimates from the Adams County data. The plot displays the distribution of ideal points among Biden (Democratic) and Trump (Republican) voters in the 2020 election. Both the MCMC and VAE plots are clearly bimodal and place Biden (Trump) voters on the left (right) side of the axis. This matches the findings from other research in this area.[6] In the MCMC models, the two clear peaks around 0 and 0.7 are primarily straight-ticket voters for Democratic and Republican candidates respectively. For those voters, it can be hard to ascertain exactly where they are in the space relative to other strong party voters, a common criticism of ideal point models. In the right plot of Figure 2, I directly plot the estimates from each model against one another. Again, there is clear separation between the two parties, and the estimated ideal points correlate at 0.95. As can be seen in the plot, the models fit very similarly to one another.

The speed of estimation allows me to fit a model to votes from the entire nation and jointly scale all voters. The bridging contest in this case is only the U.S. President, but initial results suggest this is sufficient.[7] I fit a

[6] For example, Bafumi and Herron (2010) plots a roughly bimodal, although largely inseperable, distribution of voter ideal points measured using the 2006 CCES in Figure 5.

[7] There are other contests withiin states and within districts that bridge individual voters to one another, but the only completely unifying contest is the U.S. President.

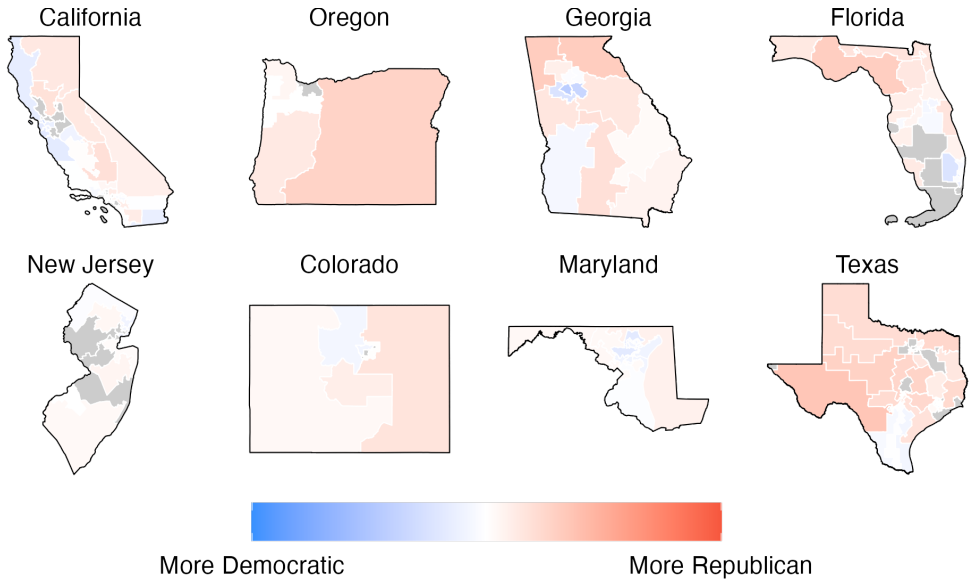

Figure 3: Underlying Ideal Point Estimates in U.S. House Districts. The plot displays the mean ideal point among voters in each House district in the U.S. Districts that are more blue lean more towards the Democrats, whereas districts that are more red lean towards the Republicans. Gray districts can represent either uncontested elections or districts where entire counties are missing from the CVR data.

one-dimensional IRT model, then plot the underlying ideal point distribution for every U.S. House district in Figure 3. Facially, the results seem to line up with my expectation, where more rural districts are more Republican and conservative than urban districts. In Figure B.3 I plot the ideal points for each county in the data, mirroring Figure B.1.

One argument against estimating ideal point estimates is that they are no better than simply using the election results and calculating the Democratic (Republican) share of the vote. How do the ideal point estimates compare to simple vote totals? Using the VAE estimates, in Figure 4, I take the mean ideal point in various Colorado State House districts compared to the two-party vote share for Democrats from precinct-level election results (Baltz et al., 2022). The left panels show the entire state while the right panel includes a zoomed inset of the Denver metro area. Setting aside uncontested elections and house districts with no CVR data (represented with grey districts), there are some interesting differences in the plots. For example, despite electing a Democrat in a southwestern district (District 62) of Colorado, the underlying ideal point model suggests that the district is almost perfectly split (represented by the white background). Donald E. Valdez was the Democratic candidate in District 62, who received 57.8% of the vote compared with the 51.7% Joe Biden received from the same voters. This could suggest that Valdez is a popular candidate among both parties who gets votes from people who are otherwise Republican voters. Other districts in Colorado also exhibit the same pattern of underlying ideological disagreement. Formally, Figure B.2 compares ideological estimates from the VAE model to two-party vote share estimates for the president. The two models generally agree, with a correlation of 0.93, but are not totally in agreement, especially near the tails where the ideal point estimates tend to indicate more moderation than the election results.

4.3 Representation in State Legislatures

Recent work assesses representation in state legislatures, mostly painting a bleak picture (Rogers, 2017, 2023). Rogers compares survey-based estimates of district-level ideology from Tausanovitch and Warshaw (2013) with estimates of the ideology of state legislators from Shor and McCarty (2011). Tausanovitch and Warshaw

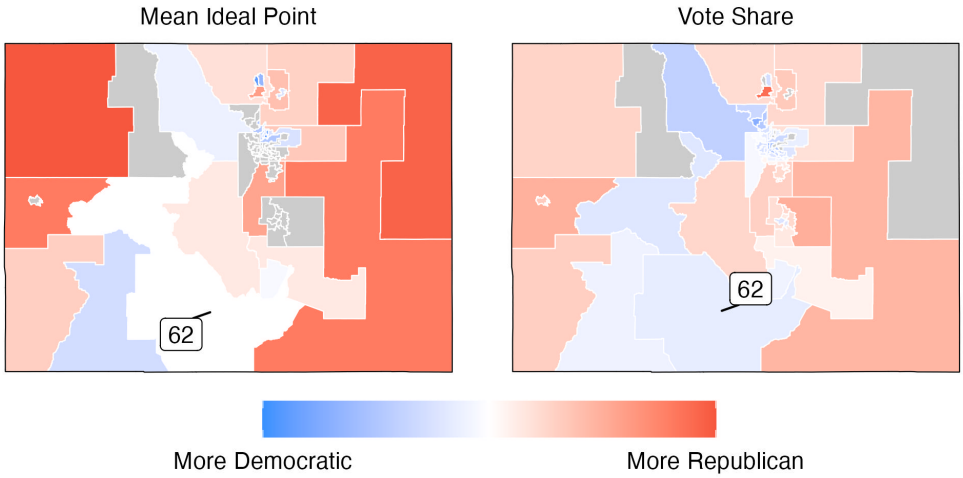

Figure 4: Comparison of Vote Share in State House Elections and Underlying Ideal Point Estimates. The left plot displays the mean ideal point among voters in each state house district in Colorado. The right plot displays the same districts, but 'ideology' is instead measured as the two-party margin in the state house election, using precinct-level results (Baltz et al., 2022). Districts that are more blue lean more towards the Democrats, whereas districts that are more red lean towards the Republicans. Both plots are drawn on a common Democratic–Republican scale and share a single legend: the mean ideal points are rescaled so that the median district in the state falls at the midpoint of the scale, matching the midpoint of the vote margin. District 62, which I discuss in the text, is labeled in both plots. In the right plot, gray districts represent uncontested elections where a vote share breakdown does not make sense. In the left plot, gray districts can represent either uncontested elections (the exact same elections as in the right plot) or districts where entire counties are missing from the CVR data. A clear example of the missing counties is the cluster of gray districts immediately southeast of Denver – these are part of Arapahoe County, a county which is missing valid data in the CVR dataset. See Kuriwaki et al. (2024) for more information on what defines "valid" data in this context.

(2013) estimates are constructed from survey estimates and Shor and McCarty (2011) estimates are constructed from roll-call votes. They both have their own advantages and disadvantages, but here I focus on providing an alternative, ballot based estimate of district ideology. My method is more faithful to the true ballot a voter faces and has a much larger sample size, but it is also necessarily constrained by the possible choices on a ballot each voter could have cast a ballot for.

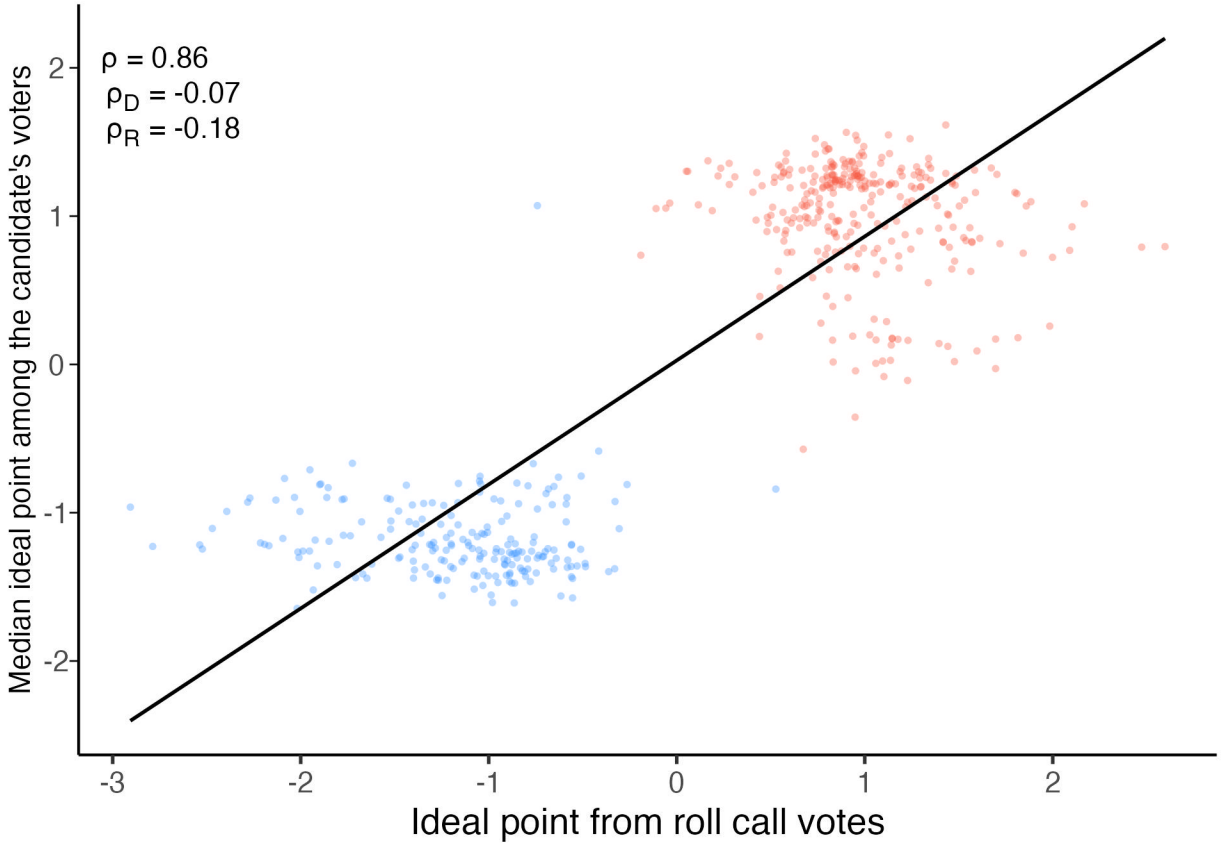

Figure 5: Comparison of the median voter from ideal point estimates for each state legislative candidate to their roll-call vote based estimate. Each point in the plot represents one candidate in a state legislative election with both a roll-call based ideal point (Shor and McCarty, 2011) and a CVR based ideal point (this paper). Blue points indicate Democratic candidates and red points indicate Republican candidates. The black line is the 45-degree line. In the top left are correlations for all points together, then just for Democrats (ρ_D) and just for Republicans (ρ_R).

5 Conclusion

Estimation of ideal points is a common statistical task in political science, but the increasing size of datasets has made traditional methods for estimating ideal points computationally infeasible. This paper has introduced a new method for estimating ideal points using variational autoencoders (VAEs). VAEs are a type of neural network that can be used to estimate latent variables in large datasets. I have shown that VAEs can be used to estimate ideal points from large datasets orders of magnitudes faster than traditional methods. This type of model also nicely balances the tradeoff between interpretability and flexibility, allowing for the estimation of multidimensional ideal points while still maintaining a clear IRT interpretation. As the types of data researchers use continues to grow in scale and complexity, VAEs are a promising new tool for estimating

ideal points in political science.

References

Abi-Hassan, S., Box-Steffensmeier, J. M., Christenson, D. P., Kaufman, A. R., and Libgober, B. (2023). The Ideologies of Organized Interests and Amicus Curiae Briefs: Large-Scale, Social Network Imputation of Ideal Points. *Political Analysis*, 31(3):396–413.

Alvarez, R. M., Hall, T. E., and Levin, I. (2018). Low-Information Voting: Evidence From Instant-Runoff Elections. *American Politics Research*, 46(6):1012–1038.

Bafumi, J. and Herron, M. C. (2010). Leapfrog Representation and Extremism: A Study of American Voters and Their Members in Congress. *American Political Science Review*, 104(3):519–542.

Bafumi, J., Herron, M. C., Hill, S. J., and Lewis, J. B. (2012). Alvin Greene? Who? How Did He Win the United States Senate Nomination in South Carolina? *Election Law Journal: Rules, Politics, and Policy*, 11(4):358–379.

Bailey, M. A., Strezhnev, A., and Voeten, E. (2017). Estimating Dynamic State Preferences from United Nations Voting Data. *Journal of Conflict Resolution*, 61(2):430–456.

Baltz, S., Agadjanian, A., Chin, D., Curiel, J., DeLuca, K., Dunham, J., Miranda, J., Phillips, C. H., Uhlman, A., and Wimpy, C. (2022). American election results at the precinct level. *Scientific Data*, 9(1):651.

Barberá, P. (2015). Birds of the same feather tweet together: Bayesian ideal point estimation using Twitter data. *Political analysis*, 23(1):76–91.

Bond, R. and Messing, S. (2015). Quantifying social media's political space: Estimating ideology from publicly revealed preferences on Facebook. *American Political Science Review*, 109(1):62–78.

Bonica, A. (2024). Database on Ideology, Money in Politics, and Elections: Public Version 4.0. Place: Stanford, CA.

Burnham, M. (2024). Semantic Scaling: Bayesian Ideal Point Estimates with Large Language Models. Working paper. https://arxiv.org/abs/2405.02472v1.

Chalmers, R. P. (2012). **mirt** : A Multidimensional Item Response Theory Package for the *R* Environment. *Journal of Statistical Software*, 48(6).

Cho, A. E., Wang, C., Zhang, X., and Xu, G. (2021). Gaussian variational estimation for multidimensional item response theory. *British Journal of Mathematical and Statistical Psychology*, 74(S1):52–85.

Collier, M., Nazabal, A., and Williams, C. K. I. (2020). VAEs in the Presence of Missing Data. Working paper. https://arxiv.org/abs/2006.05301v3.

Conevska, A. and Mutlu, C. (2025). When Do Voters Stop Caring? Estimating the Shape of Voter Utility Function. Working paper. http://arxiv.org/abs/2501.03196.

Converse, G., Curi, M., Oliveira, S., and Templin, J. (2021). Estimation of multidimensional item response theory models with correlated latent variables using variational autoencoders. *Machine Learning*, 110(6):1463–1480.

Cowburn, M. and Sältzer, M. (2025). Partisan communication in two-stage elections: the effect of primaries on intra-campaign positional shifts in congressional elections. *Political Science Research and Methods*, 13(2):392–411.

Curi, M., Converse, G. A., Hajewski, J., and Oliveira, S. (2019). Interpretable Variational Autoencoders for Cognitive Models. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Dowling, C. M., Miller, M. G., and Morris, K. T. (2025). Crossover Voting Rates in Partisan and Nonpartisan Elections: Evidence From Cast Vote Records. *Political Research Quarterly*, 78(2):720–737.

Däubler, T. and Benoit, K. (2017). Estimating better left-right positions through statistical scaling of manual content analysis.

Egami, N., Hinck, M., Stewart, B. M., and Wei, H. (2024). Using large language model annotations for the social sciences: A general framework of using predicted variables in downstream analyses. Working paper. https://naokiegami.com/paper/dsl_ss.pdf.

Enelow, J. M. and Hinich, M. J. (1984). *The spatial theory of voting: An introduction*. Cambridge University Press, Cambridge.

Frisina, L., Herron, M. C., Honaker, J., and Lewis, J. B. (2008). Ballot Formats, Touchscreens, and Undervotes: A Study of the 2006 Midterm Elections in Florida. *Election Law Journal: Rules, Politics, and Policy*, 7(1):25–47.

Gerber, E. and Lewis, J. (2004). Beyond the Median: Voter Preferences, District Heterogeneity, and Political Representation. *Journal of Political Economy*, 112(6):1364–1383.

Grimmer, J. (2011). An Introduction to Bayesian Inference via Variational Approximations. *Political Analysis*, 19(1):32–47.

Guess, A. M., Nyhan, B., and Reifler, J. (2020). Exposure to untrustworthy websites in the 2016 US election. *Nature Human Behaviour*, 4(5):472–480.

Herrmann, M. and Döring, H. (2023). Party Positions from Wikipedia Classifications of Party Ideology. *Political Analysis*, 31(1):22–41.

Herron, M. C. and Lewis, J. B. (2007). Did Ralph Nader spoil Al Gore's presidential bid? A ballot-level study of green and reform party voters in the 2000 presidential election. *Quarterly Journal of Political Science*, 2(3):205–226.

Herron, M. C. and Sekhon, J. S. (2003). Overvoting and representation: An examination of overvoted presidential ballots in Broward and Miami-Dade Counties. *Electoral Studies*, 22(1):21–47.

Imai, K., Lo, J., and Olmsted, J. (2016). Fast Estimation of Ideal Points with Massive Data. *American Political Science Review*, 110(4):631–656.

Jackman, S. (2009). *Bayesian Analysis for the Social Sciences*. Wiley, Hoboken, NJ.

Kingma, D. P. and Welling, M. (2019). An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.

Kuriwaki, S. (2026). Ticket Splitting in a Nationalized Era. *The Journal of Politics*, 88(1):47–62.

Kuriwaki, S., Reece, M., Baltz, S., Conevska, A., Loffredo, J. R., Mutlu, C., Samarth, T., Acevedo Jetter, K. E., Djanogly Garai, Z., Murray, K., Hirano, S., Lewis, J. B., Snyder, J. M., and Stewart, C. (2024). Cast vote records: A database of ballots from the 2020 U.S. Election. *Scientific Data*, 11(1):1304.

Lai, A., Brown, M. A., Bisbee, J., Tucker, J. A., Nagler, J., and Bonneau, R. (2024). Estimating the Ideology of Political YouTube Videos. *Political Analysis*, 32(3):345–360.

Lauderdale, B. E. and Herzog, A. (2016a). Measuring political positions from legislative speech. *Political Analysis*, 24(3):374–394.

Lauderdale, B. E. and Herzog, A. (2016b). Measuring political positions from legislative speech. *Political Analysis*, 24(3):374–394.

Laver, M., Benoit, K., and Garry, J. (2003). Extracting policy positions from political texts using words as data. *American political science review*, 97(2):311–331.

Lewis, J. B. (2001). Estimating Voter Preference Distributions from Individual-Level Voting Data. *Political Analysis*, 9(3):275–297.

Lewis, J. B., Poole, K., Rosenthal, H., Boche, A., Rudkin, A., and Sonnet, L. (2025). Voteview: Congressional Roll-Call Votes Database.

Ma, C., Ouyang, J., Wang, C., and Xu, G. (2024). A Note on Improving Variational Estimation for Multidimensional Item Response Theory. *Psychometrika*, 89(1):172–204.

Martin, A. D. and Quinn, K. M. (2002). Dynamic ideal point estimation via Markov chain Monte Carlo for the US Supreme Court, 1953–1999. *Political analysis*, 10(2):134–153.

Morse, M. (2021). The Future of Felon Disenfranchisement Reform. *California Law Review*, 109(3):1143–1197.

Morucci, M., Foster, M. J., Webster, K., Lee, S. J., and Siegel, D. A. (2024). Measurement That Matches Theory: Theory-Driven Identification in Item Response Theory Models. *American Political Science Review*, 119(2):727–745.

Peress, M. (2022). Large-Scale Ideal Point Estimation. *Political Analysis*, 30(3):346–363.

Peterson, A. and Spirling, A. (2018). Classification accuracy as a substantive quantity of interest: Measuring polarization in westminster systems. *Political Analysis*, 26(1):120–128.

Poole, K. T. and Rosenthal, H. (1985). A spatial model for legislative roll call analysis. *American journal of political science*, 29(2):357–384.

Reece, M., Péloquin-Skulski, G., Murray, K., Loffredo, J., Acevedo Jetter, K. E., Gonzalez, F., Garai, Z. D., Flores, A., Braculj, L. B., Baltz, S., and Stewart III, C. (2024). Hidden Partisanship in American Elections. Working paper, MIT Political Science Department Research Paper. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4721012.

Rivers, D. (2003). Identification of Multidimensional Item Response Models. Working paper.

Rogers, S. (2017). Electoral accountability for state legislative roll calls and ideological representation. *American Political Science Review*, 111(3):555–571.

Rogers, S. (2023). *Accountability in state legislatures*. University of Chicago Press.

Shin, S. (2025). Measuring Issue Specific Ideal Points from Roll Call Votes. Working paper. https://sooahnshin.com/issueirt.pdf.

Shor, B. and McCarty, N. (2011). The ideological mapping of American legislatures. *American Political Science Review*, 105(3):530–551.

Slapin, J. B. and Proksch, S. (2008). A Scaling Model for Estimating Time-Series Party Positions from Texts. *American Journal of Political Science*, 52(3):705–722.

Stan Development Team (2026). Stan Reference Manual.

Stier, S., Kirkizh, N., Froio, C., and Schroeder, R. (2020). Populist Attitudes and Selective Exposure to Online News: A Cross-Country Analysis Combining Web Tracking and Surveys. *The International Journal of Press/Politics*, 25(3):426–446.

Tai, Y. C., Nakka, N., Patni, K. N., Rajtmajer, S., Munger, K., Lin, Y.-R., and Desmarais, B. A. (2025). The digitally accountable public representation database: online communication by U.S. officials. *Scientific Data*, 12(1):1556.

Tausanovitch, C. and Warshaw, C. (2013). Measuring Constituent Policy Preferences in Congress, State Legislatures, and Cities. *The Journal of Politics*, 75(2):330–342.

Urban, C. J. and Bauer, D. J. (2021). A deep learning algorithm for high-dimensional exploratory item factor analysis. *Psychometrika*, 86(1):1–29.

Veldkamp, K., Grasman, R., and Molenaar, D. (2025). Handling missing data in variational autoencoder based item response theory. *British Journal of Mathematical and Statistical Psychology*, 78(1):378–397.

Wand, J. N., Shotts, K. W., Sekhon, J. S., Mebane Jr, W. R., Herron, M. C., and Brady, H. E. (2001). The butterfly did it: The aberrant vote for Buchanan in Palm Beach County, Florida. *American Political Science Review*, 95(4):793–810.

Wu, M., Davis, R. L., Domingue, B. W., Piech, C., and Goodman, N. (2020). Variational Item Response Theory: Fast, Accurate, and Expressive. Working paper. http://arxiv.org/abs/2002.00276.

Yamamoto, T. (2014). A Multinomial Response Model for Varying Choice Sets, with Application to Partially Contested Multiparty Elections. Working paper.

Zhang, L., Carpenter, B., Gelman, A., and Vehtari, A. (2022). Pathfinder: Parallel quasi-Newton variational inference. *Journal of Machine Learning Research*, 23(306):1–49.

Appendix

Intended for online publication only.

Table of contents

A Proofs

A.1 Mathematical Comparison to MCMC, EM, and Variational Inference

All four methods share a common inferential target: characterize the posterior $p(\boldsymbol{\alpha}, \boldsymbol{\Omega} \mid \mathbf{Y})$, or equivalently, maximize the marginal log-likelihood

$$\log p(\mathbf{Y}; \boldsymbol{\Omega}) = \log \int p(\mathbf{Y} \mid \boldsymbol{\alpha}; \boldsymbol{\Omega})\, p(\boldsymbol{\alpha})\, d\boldsymbol{\alpha}.$$

This integral is intractable in closed form for the IRT likelihood in Eq. 2, so each method handles the intractability differently — with meaningful consequences for computational cost.

MCMC. Hamiltonian Monte Carlo and related samplers draw correlated samples $(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\Omega}^{(t)})$ from the joint posterior by simulating Hamiltonian dynamics in the parameter space. The proposal uses the gradient of the log-posterior

$$\nabla \log p(\boldsymbol{\alpha}, \boldsymbol{\Omega} \mid \mathbf{Y}) = \sum_{j=1}^{N} \nabla \log L(\boldsymbol{\Omega}, \alpha_j \mid \mathbf{y}_j) + \nabla \log p(\boldsymbol{\alpha}) + \nabla \log p(\boldsymbol{\Omega}),$$

which requires a full pass over all N voters and K contests at every leapfrog step. MCMC is the only method here that is asymptotically exact — given enough samples, the empirical distribution of the chain converges to the true posterior. However, the parameter space has dimension $O(N + K)$, so both per-iteration cost and mixing time grow with the number of voters, making MCMC prohibitively slow for large N.

EM. Expectation-maximization treats the voter ideal points $\boldsymbol{\alpha}$ as latent variables and targets the marginal likelihood $p(\mathbf{Y}; \boldsymbol{\Omega})$ by iterating between:

- *E-step:* Compute $Q(\boldsymbol{\Omega} \mid \boldsymbol{\Omega}^{(t)}) = \mathbb{E}_{p(\boldsymbol{\alpha} \mid \mathbf{Y}; \boldsymbol{\Omega}^{(t)})}[\log p(\mathbf{Y}, \boldsymbol{\alpha};\ \boldsymbol{\Omega})]$.
- *M-step:* $\boldsymbol{\Omega}^{(t+1)} = \operatorname{argmax}_{\boldsymbol{\Omega}}\ Q(\boldsymbol{\Omega} \mid \boldsymbol{\Omega}^{(t)})$.

The E-step requires the posterior expectation $\mathbb{E}[\alpha_j \mid \mathbf{y}_j; \boldsymbol{\Omega}^{(t)}]$ for each voter separately. Because the IRT likelihood is non-conjugate, this integral lacks a closed form and is typically approximated with Gauss–Hermite quadrature using G points per latent dimension D, giving a per-iteration cost of $O(NKG^D)$. EM converges to a local maximum of the marginal likelihood and does not produce uncertainty estimates for $\boldsymbol{\alpha}$ or $\boldsymbol{\Omega}$ without additional computation. Cost also scales linearly in N, though constant factors are smaller than MCMC because the M-step for $\boldsymbol{\Omega}$ can exploit structure in the IRT likelihood (Imai et al., 2016; Peress, 2022).

Variational inference. VI reframes posterior inference as optimization. For a chosen variational family $\mathcal{Q}$, it seeks

$$q^*(\boldsymbol{\alpha}) = \operatorname{argmin}_{q \in \mathcal{Q}}\ \mathrm{KL}[q(\boldsymbol{\alpha}) \,\|\, p(\boldsymbol{\alpha} \mid \mathbf{Y}; \boldsymbol{\Omega})]\,,$$

which is equivalent to maximizing the ELBO in Eq. 3. The standard mean-field choice $q(\boldsymbol{\alpha}) = \prod_j q_j(\alpha_j)$ with Gaussian factors introduces $2ND$ free variational parameters — a pair (μ_j, σ_j) per latent dimension per voter. Optimizing these parameters via gradient ascent costs $O(NK)$ per step and requires storing and updating $O(ND)$ parameters. Memory and computation therefore still scale linearly in N; the gain over MCMC comes from replacing sampling with gradient steps, which can exploit automatic differentiation and GPU parallelism, and from restricting attention to the ELBO rather than the full joint distribution.

VAE (amortized VI). The VAE replaces the $O(ND)$ per-voter variational parameters with a single encoder network f_ϕ shared across all voters:

$$(\mu_j, \sigma_j) = f_\phi(\mathbf{y}_j), \qquad \boldsymbol{\alpha}_j \sim q_\phi(\boldsymbol{\alpha}_j \mid \mathbf{y}_j) = \mathcal{N}(\mu_j, \operatorname{diag}(\sigma_j^2)),$$

as stated in Eq. 5. The objective is the same ELBO as in Eq. 6, but summed over a mini-batch of size $B \ll N$:

$$\widehat{\text{ELBO}}_\phi = \frac{1}{B} \sum_{j \in \mathcal{B}} \Big[\mathbb{E}_{q_\phi(\boldsymbol{\alpha}_j \mid \mathbf{y}_j)} \big[\log p(\mathbf{y}_j \mid \boldsymbol{\alpha}_j; \Omega) \big] - \text{KL}\big[q_\phi(\boldsymbol{\alpha}_j \mid \mathbf{y}_j) \,\|\, p(\boldsymbol{\alpha}_j) \big] \Big].$$

The reparameterization in Eq. 8 enables unbiased, low-variance gradient estimates of this objective with respect to ϕ. Crucially, the number of parameters to optimize is $|\phi| + |\Omega|$, which is determined by the network architecture and the number of contests — not by the number of voters. Each gradient step costs $O(BK)$, independent of N, allowing training to be distributed across GPU cores and mini-batches to be drawn without loading the full dataset into memory.

The amortization comes at a cost: the encoder must share a single set of parameters ϕ across all voters, so it cannot perfectly tailor the approximation $q_\phi(\boldsymbol{\alpha}_j \mid \mathbf{y}_j)$ to each voter the way mean-field VI can. In practice, the gap is small when the encoder is sufficiently expressive and the data are large, and the computational savings far outweigh it for the applications considered here.

Table A.1 summarizes the key properties of each method.

Table A.1: Comparison of ideal point estimation methods. N = voters, K = contests, D = latent dimensions, G = quadrature points per dimension, B = mini-batch size.

Method	Variational parameters	Per-step cost	Scales with N?	Asymptotically exact?	Uncertainty for $\boldsymbol{\alpha}$?
MCMC	$\boldsymbol{\alpha}$, Ω (sampled)	$O(NK)$	Yes	Yes	Yes
EM	Ω (marginalize $\boldsymbol{\alpha}$)	$O(NKG^D)$	Yes	No (local max)	No
VI	$\{(\mu_j, \sigma_j)\}_{j=1}^N$, Ω	$O(NK)$	Yes	No (approx.)	Yes (approx.)
VAE	ϕ, Ω (shared encoder)	$O(BK)$	No	No (approx.)	Yes (approx.)

A.2 Derivation of the ELBO for VAEs

The marginal log-likelihood of observations $\mathbf{y}$ requires integrating out the latent ideal points,

$$\log p(\mathbf{y}) = \log \int p(\mathbf{y}|\boldsymbol{\alpha}; \Omega)\, p(\boldsymbol{\alpha})\, d\boldsymbol{\alpha}, \tag{13}$$

which is intractable for nonlinear decoders. I introduce a variational distribution $q_\phi(\boldsymbol{\alpha}|\mathbf{y})$ by multiplying and dividing inside the logarithm,

$$\log p(\mathbf{y}) = \log \int p(\mathbf{y}|\boldsymbol{\alpha}; \Omega)\, p(\boldsymbol{\alpha})\, \frac{q_\phi(\boldsymbol{\alpha}|\mathbf{y})}{q_\phi(\boldsymbol{\alpha}|\mathbf{y})}\, d\boldsymbol{\alpha} = \log\, \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{y})} \left[\frac{p(\mathbf{y}|\boldsymbol{\alpha}; \Omega)\, p(\boldsymbol{\alpha})}{q_\phi(\boldsymbol{\alpha}|\mathbf{y})} \right]. \tag{14}$$

Because log is concave, Jensen's inequality gives $\log \mathbb{E}[X] \geq \mathbb{E}[\log X]$, so

$$\log p(\mathbf{y}) \geq \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{y})}\left[\log \frac{p(\mathbf{y}|\boldsymbol{\alpha};\Omega)\, p(\boldsymbol{\alpha})}{q_\phi(\boldsymbol{\alpha}|\mathbf{y})}\right] \equiv \mathcal{L}(\phi, \Omega; \mathbf{y}). \tag{15}$$

Expanding the logarithm separates the bound into a reconstruction term and a negative KL divergence,

$$\mathcal{L}(\phi, \Omega; \mathbf{y}) = \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{y})}\left[\log p(\mathbf{y}|\boldsymbol{\alpha};\Omega)\right]}_{\text{reconstruction}} - \underbrace{\mathrm{KL}\left[q_\phi(\boldsymbol{\alpha}|\mathbf{y}) \,\|\, p(\boldsymbol{\alpha})\right]}_{\text{regularization}}. \tag{16}$$

The gap between the log-likelihood and the ELBO is exactly the KL divergence between the variational posterior and the true posterior. To see this, write

$$\log p(\mathbf{y}) = \mathcal{L}(\phi, \Omega; \mathbf{y}) + \mathrm{KL}\left[q_\phi(\boldsymbol{\alpha}|\mathbf{y}) \,\|\, p(\boldsymbol{\alpha}|\mathbf{y})\right], \tag{17}$$

which follows from substituting Bayes' rule, $p(\mathbf{y}|\boldsymbol{\alpha};\Omega)\, p(\boldsymbol{\alpha}) = p(\boldsymbol{\alpha}|\mathbf{y})\, p(\mathbf{y})$, into Eq. 15: the integrand becomes $\log p(\mathbf{y}) - \log\left[q_\phi(\boldsymbol{\alpha}|\mathbf{y})/p(\boldsymbol{\alpha}|\mathbf{y})\right]$, and $\log p(\mathbf{y})$ passes outside the expectation because it does not depend on $\boldsymbol{\alpha}$. Since $\log p(\mathbf{y})$ is fixed with respect to ϕ and $\mathrm{KL}[\cdot\|\cdot] \geq 0$, maximizing $\mathcal{L}$ over ϕ is equivalent to minimizing $\mathrm{KL}[q_\phi(\boldsymbol{\alpha}|\mathbf{y})\|p(\boldsymbol{\alpha}|\mathbf{y})]$, driving the variational family toward the true posterior.

A.3 Derivation of the Reparameterization Trick

Training requires computing gradients of the ELBO with respect to the encoder parameters ϕ. The difficulty is that the reconstruction term in Eq. 16 involves an expectation whose sampling distribution itself depends on ϕ,

$$\nabla_\phi\, \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{y})}\left[\log p(\mathbf{y}|\boldsymbol{\alpha};\Omega)\right] = \nabla_\phi \int q_\phi(\boldsymbol{\alpha}|\mathbf{y}) \log p(\mathbf{y}|\boldsymbol{\alpha};\Omega)\, d\boldsymbol{\alpha}, \tag{18}$$

so the gradient cannot simply pass through the expectation — this is the inequality noted in Eq. 7 of the main text. The score-function (REINFORCE) estimator rewrites this as $\mathbb{E}_{q_\phi}[\log p(\mathbf{y}|\boldsymbol{\alpha};\Omega)\nabla_\phi \log q_\phi(\boldsymbol{\alpha}|\mathbf{y})]$, which is unbiased but has notoriously high variance due to the score term, making optimization unstable in practice.

The reparameterization trick resolves this by expressing the stochasticity through a fixed, ϕ-independent noise variable. For the Gaussian approximate posterior $q_\phi(\boldsymbol{\alpha}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_\phi, \mathrm{diag}(\boldsymbol{\sigma}_\phi^2))$, any draw $\boldsymbol{\alpha} \sim q_\phi$ can be written as

$$\boldsymbol{\alpha} = \boldsymbol{\mu}_\phi(\mathbf{y}) + \boldsymbol{\sigma}_\phi(\mathbf{y}) \odot \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{19}$$

where $\odot$ denotes element-wise multiplication. This is the transformation introduced in Eq. 8 of the main text. Under this change of variable the expectation over q_ϕ becomes an expectation over the fixed distribution $p(\boldsymbol{\varepsilon}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$,

$$\mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{y})}[f(\boldsymbol{\alpha})] = \mathbb{E}_{p(\boldsymbol{\varepsilon})}\left[f\left(\boldsymbol{\mu}_\phi + \boldsymbol{\sigma}_\phi \odot \boldsymbol{\varepsilon}\right)\right]. \tag{20}$$

Because $p(\boldsymbol{\varepsilon})$ does not depend on ϕ, the gradient operator passes through the expectation,

$$\nabla_\phi \, \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{y})}[f(\boldsymbol{\alpha})] = \mathbb{E}_{p(\boldsymbol{\varepsilon})}\Big[\nabla_\phi \, f\Big(\boldsymbol{\mu}_\phi + \boldsymbol{\sigma}_\phi \odot \boldsymbol{\varepsilon}\Big)\Big], \tag{21}$$

and standard automatic differentiation applies. In practice the expectation in Eq. 21 is estimated with S Monte Carlo draws $\boldsymbol{\varepsilon}^{(s)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $s = 1, \ldots, S$,

$$\nabla_\phi \, \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{y})}[\log p(\mathbf{y}|\boldsymbol{\alpha}; \Omega)] \approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\phi \log p\Big(\mathbf{y} \,\Big|\, \boldsymbol{\mu}_\phi + \boldsymbol{\sigma}_\phi \odot \boldsymbol{\varepsilon}^{(s)}; \Omega\Big), \tag{22}$$

where $S = 1$ per voter typically suffices for stable training, because averaging gradients over the voters in a mini-batch already reduces the variance of the estimator. The KL term of Eq. 16 requires no Monte Carlo approximation: for a Gaussian variational posterior and standard normal prior it has a closed form (given in Eq. 25 below), so its gradient with respect to ϕ is computed exactly.

A.4 Derivation of the Loss Function for Conditional VAEs

Cast vote records are inherently incomplete: a voter who did not participate in a down-ballot race provides no information about that race and voters do not participate in all races since they face different ballots in different jurisdictions. I therefore condition both the encoder and decoder on a binary missingness mask $\mathbf{m}_j \in \{0, 1\}^K$, where K is the number of races and $m_{jk} = 1$ indicates that voter j cast a vote in race k. The encoder becomes $q_\phi(\boldsymbol{\alpha}_j|\mathbf{y}_j, \mathbf{m}_j)$, and the decoder becomes $p(\mathbf{y}_j|\boldsymbol{\alpha}_j, \mathbf{m}_j; \Omega)$.

The quantity to bound is the conditional log-likelihood $\log p(\mathbf{y}_j|\mathbf{m}_j)$, and the derivation of Eq. 15 carries through with $\mathbf{m}_j$ appended as a conditioning variable throughout. This yields the masked ELBO,

$$\mathcal{L}_{\text{CVAE}}(\phi, \Omega; \mathbf{y}_j, \mathbf{m}_j) = \mathbb{E}_{q_\phi(\boldsymbol{\alpha}_j|\mathbf{y}_j, \mathbf{m}_j)}\big[\log p(\mathbf{y}_j|\boldsymbol{\alpha}_j, \mathbf{m}_j; \Omega)\big] - \mathrm{KL}\big[q_\phi(\boldsymbol{\alpha}_j|\mathbf{y}_j, \mathbf{m}_j) \,\|\, p(\boldsymbol{\alpha}_j)\big]. \tag{23}$$

The prior on ideal points is taken to be independent of the participation pattern, $p(\boldsymbol{\alpha}_j|\mathbf{m}_j) = p(\boldsymbol{\alpha}_j) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, a standard modeling choice for CVAEs.

Assuming ballot choices are conditionally independent across races given $\boldsymbol{\alpha}_j$, and noting that $\mathbf{y}_j$ contains entries only for races in which voter j participated, the log-likelihood of the observed choices reduces to a sum over observed races,

$$\log p(\mathbf{y}_j|\boldsymbol{\alpha}_j, \mathbf{m}_j; \Omega) = \sum_{k=1}^{K} m_{jk} \log p(y_{jk}|\boldsymbol{\alpha}_j; \Omega), \tag{24}$$

so the reconstruction loss sums only over the $m_{jk} = 1$ entries and unobserved races contribute nothing to the gradient. The mask enters the decoder only through this gating: conditional on participation, the categorical probabilities for race k depend on $\boldsymbol{\alpha}_j$ alone, which is why the per-race terms are written $p(y_{jk}|\boldsymbol{\alpha}_j; \Omega)$. This treats $\mathbf{m}_j$ as conditioning information rather than as data to be modeled; the generative model is agnostic about why a race is unobserved. For each observed race k the likelihood is categorical with probabilities given by a softmax over the decoder's logits, so this term evaluates to the masked cross-entropy.

For the KL term, the prior is $p(\boldsymbol{\alpha}_j) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the approximate posterior is $q_\phi(\boldsymbol{\alpha}_j|\mathbf{y}_j, \mathbf{m}_j) = \mathcal{N}(\boldsymbol{\mu}_j, \mathrm{diag}(\boldsymbol{\sigma}_j^2))$. The KL divergence between two Gaussians has a closed form that requires no Monte Carlo estimation,

$$\mathrm{KL}\Big[\mathcal{N}(\boldsymbol{\mu}_j, \mathrm{diag}(\boldsymbol{\sigma}_j^2)) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\Big] = \frac{1}{2}\sum_{d=1}^{D}\left(\sigma_{jd}^2 + \mu_{jd}^2 - 1 - \log\sigma_{jd}^2\right). \tag{25}$$

Combining Eq. 24 and Eq. 25, approximating the reconstruction expectation with a single reparameterized draw as in Eq. 22, and negating (since optimizers minimize), the training loss for voter j is

$$\mathcal{J}(\phi, \Omega; \mathbf{y}_j, \mathbf{m}_j) = -\sum_{k=1}^{K} m_{jk} \log p(y_{jk} | \boldsymbol{\alpha}_j^{(s)}; \Omega) + \frac{1}{2}\sum_{d=1}^{D}\left(\sigma_{jd}^2 + \mu_{jd}^2 - 1 - \log\sigma_{jd}^2\right), \tag{26}$$

where $\boldsymbol{\alpha}_j^{(s)} = \boldsymbol{\mu}_j + \boldsymbol{\sigma}_j \odot \boldsymbol{\varepsilon}^{(s)}$ with $\boldsymbol{\varepsilon}^{(s)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a reparameterized sample. The full training objective sums $\mathcal{J}$ over all voters $j = 1, \ldots, N$ and is minimized via stochastic gradient descent on mini-batches. The mask $\mathbf{m}_j$ thus plays a dual role: it gates the reconstruction loss so that missing races are ignored, and it is passed as input to the encoder so the network can distinguish abstention from a low-probability vote.

Table B.2: Representativeness of CVR sample counties vs. all U.S. counties (2020). Population-weighted county means. Median-age weighting uses county medians. Alaska and Delaware only report their data at the state level and are treated as a single county for comparison purposes.

	CVR sample	All U.S. counties
County characteristics		
% White (non-Hispanic)	52.5	60.1
% Black	12.6	12.2
% Hispanic	24.7	18.2
% Asian	6.5	5.6
Median age	38.2	38.6
% age 65+	15.8	16.0
% bachelor's degree or higher	33.4	32.9
% rural	13.1	20.2
2020 Dem. two-party vote share	53.6	52.5
Sample coverage		
Counties	289	3,112
Population (millions)	55.6	326.6

B Additional Details on Empirical Application

B.1 Cast Vote Record Data

I use ballot-level data, commonly called "cast vote records" (CVRs), from the 2020 general election. CVRs are anonymized records of all the choices made by each voter on their ballot. The data is fully anonymous and I am unable to identify anything about the voters except where they voted. The data has been standardized into a common format across the nation to facilitate analysis and has been carefully validated to ensure the data match aggregated election results.[8]

CVRs from the 2020 election were released by some local election administrators in response to increasing calls from election activists to be able to independently verify the results of the 2020 election.[9] The data contains over 594.3 million choices in elections at all levels of government and in localities of all kinds from all over the country, representing the votes of 34.8 million voters. A comparison of the demographics of the CVR counties to all counties in the nation can be found in Table B.2. I first remove all candidates who received less than 50 votes in the data, then remove all races where only one candidate was contesting the election, and finally remove all races where a voter could select more than one candidate.[10] The full set of CVRs covers 17 states, 289 counties, 34.8 million voters and 7,164 unique contests. See Figure B.1 for the full distribution of the data. This data allow me to simultaneously study local-level heterogeneity and make comparisons between jurisdictions.

B.2 MCMC Model Architecture

To validate the VAE estimates I fit a fully Bayesian version of the same categorical two-parameter IRT model described in Eq. 1 using Hamiltonian Monte Carlo (HMC) as implemented in Stan (Stan Development Team,

[8] More details on this process can be found in Kuriwaki et al. (2024).

[9] Some states prohibit releasing CVRs in any form, and for other states the ability to do so depended on whether a group requested them from a certain county, whether the county used technology enabling this data to be easily compiled, and whether the county election office had the capacity to even complete the request.

[10] Future research should model these types of races since they represent about 10% of all elections in the data.

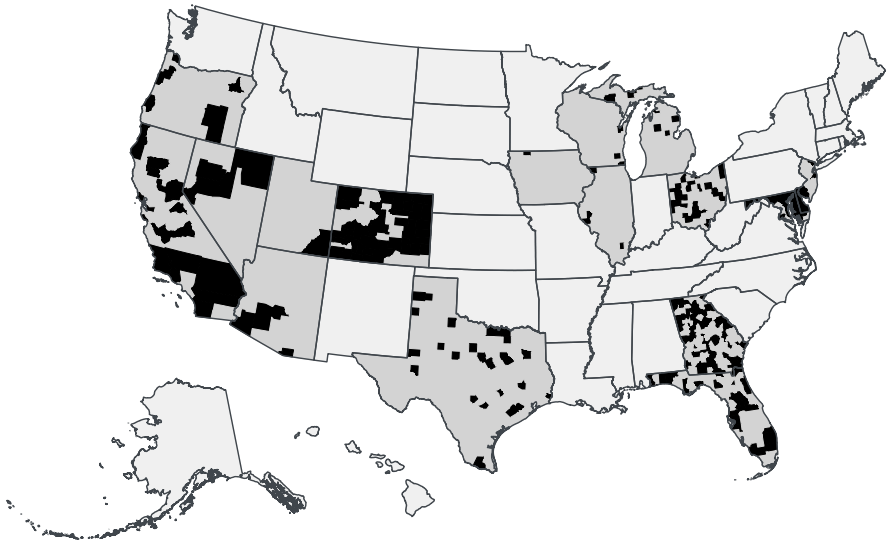

Figure B.1: Cast Vote Record Distribution. This is a map of the United States subdivided by state and county. Present counties are black, gray counties are in a state with some representation in the data, and light gray counties are in a state with no representation in the data.

2026). Because HMC does not scale to the full dataset, I estimate this benchmark model on a single county, Adams County, Colorado, restricted to single-magnitude contested races. This yields a sample of 12,006 voters, which is small enough to sample from directly but still large enough to provide an informative comparison. The identical set of ballots is used to compute the VAE-vs-MCMC correlations reported in Appendix C.

The likelihood is exactly Eq. 1: for voter j in contest k, the probability of choosing candidate c is the softmax of the linear predictors $\eta_{jkc} = \alpha_j \gamma_{kc} - \beta_{kc}$, where α_j is the voter's one-dimensional ideal point, γ_{kc} is the discrimination of candidate c, and β_{kc} is that candidate's difficulty. Within each contest one candidate serves as the reference category, with its discrimination and difficulty fixed to $\gamma_{kc} = \beta_{kc} = 0$; this resolves the softmax's additive indeterminacy. The remaining rotational, reflection, and scale indeterminacies of the latent space are handled through the priors below and by post-processing the draws (rescaling to unit variance and orienting the scale so that higher values correspond to more conservative voters) before comparison.

I place the following priors on the parameters. For identification, each voter's ideal point is given a standard normal prior, $\alpha_j \sim \mathcal{N}(0, 1)$, which fixes the scale and location of the latent space. The candidate-level item parameters are modeled hierarchically. Discriminations are drawn from a common normal distribution, $\gamma_{kc} \sim \mathcal{N}(0, \sigma_\gamma)$, and difficulties from a normal distribution centered on a shared mean, $\beta_{kc} \sim \mathcal{N}(\mu_\beta, \sigma_\beta)$, so that information is pooled across candidates. The hyperparameters receive weakly informative Student-t priors, $\mu_\beta \sim t_3(0, 2.5)$ and $\sigma_\gamma, \sigma_\beta \sim t_3^+(0, 2.5)$, with the scale parameters constrained to be positive. To keep the likelihood efficient, contests are stored in a ragged (long) format and the per-contest categorical likelihood is evaluated with Stan's OpenCL acceleration on a GPU.

Rather than initializing HMC from random or default starting values, I first fit the model with variational inference and use the resulting approximate posterior to initialize the sampler. Specifically, I run the Pathfinder algorithm (Zhang et al., 2022), a parallel quasi-Newton variational method, on the same data and model, and pass its draws as initial values for the Markov chains. This warm start places the chains in a region of high posterior mass, which substantially reduces warmup time and improves mixing for a model of this

dimensionality. I then run HMC with the No-U-Turn Sampler using four chains of 1,000 warmup and 1,000 sampling iterations each, for a total of 4,000 posterior draws. Convergence is assessed with the standard $\hat{R}$ and effective-sample-size diagnostics. The model converges well. The voter ideal-point estimates used for validation are the posterior means of α_j.

B.3 VAE Model Architecture

The main model reported in the paper is fit on the full pooled dataset (`combined.parquet`). Following the hyperparameter recommendations in the main text, I fit the VAE with a single hidden layer of size 128, one latent dimension, an embedding dimension of 16, mini-batches of size 1{,}024, a learning rate of 5×10^{-5}, and one importance-weighted ELBO sample. Since each contest can contain more than two candidates, I fit a categorical model with the reference candidate set either to the Democratic option or, in nonpartisan contests, randomly. I also implement the missing-data mask described in the extensions section.

The encoder is embedding-based rather than a dense network over the raw ballot. Each item (contest) has its own embedding table that maps the chosen candidate to a 16-dimensional vector; a per-item linear projection with an ELU nonlinearity is applied, the resulting vectors are averaged across only the contests a voter actually appears in (a masked mean-pool that handles the sparse, ragged ballot structure), and this pooled representation is passed through the size-128 hidden layer (again with an ELU) before two linear heads produce the mean and log–standard-deviation of the one-dimensional latent. The reparameterized latent draws are passed through a learned Cholesky transform. The decoder is deliberately shallow: it is a single linear map from the latent space to per-contest categorical logits, so that the decoder weights and biases recover the 2PL-style discrimination and difficulty parameters directly (Eq. 1) rather than being entangled in additional hidden layers. Optimization uses Adam (with AMSGrad) and early stopping on the training loss, for up to 20 epochs.

B.4 Comparison of Ideal Points to Two-Party Vote Share

B.5 Full County Map of Ideal Points

C Hyperparameter Tuning

The choice of hyperparameters for the VAE model is important for the performance of the model. I conduct a hyperparameter tuning process to evaluate the sensitivity of the model to different hyperparameter choices. The hyperparameters I tune include the batch size, hidden layer size, number of latent dimensions, number of ELBO samples, learning rate, and embedding dimension. The Bayesian MCMC estimates come from a model fit on a sample of voters in Adams County, Colorado only but the VAE estimates are performed on the full state of Colorado to maximize the accuracy. The results from the full state are then filtered down to only the ballots used for estimation in the MCMC model for comparison. Notice that unlike the main paper, which uses the fit from the entire data set, here I only use Colorado data to ensure I can reasonably iterate over many hundreds of hyperparameter combinations.

For each hyperparameter, I vary its value while keeping the other hyperparameters fixed at their default values. I then evaluate the correlation between the VAE estimates and the MCMC estimates for each iteration of the tuning process. The results of the hyperparameter tuning process are displayed in Figure C.4. The plot shows that the correlation between the VAE estimates and the MCMC estimates is relatively stable across different hyperparameter values, with some variation depending on the specific hyperparameter. Overall, the model appears to be robust to different hyperparameter choices and the default values used in the main analysis are reasonable choices for this application. I confirm this visual assumption with a simple multivariate

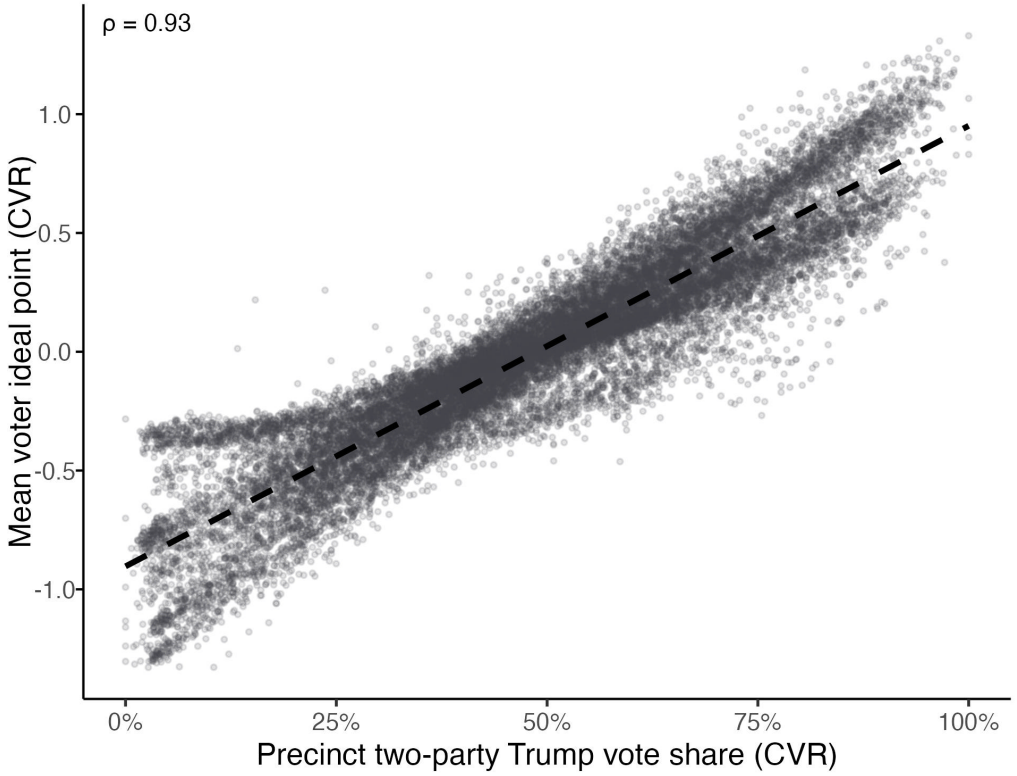

Figure B.2: Ideal point estimates of ideology versus two-party vote shares. The plot displays, for each precinct in the CVR data, the mean ideal point in the precinct matched to the two-party vote share for Trump from precinct-level election returns (Baltz et al., 2022). In the top left corner of the plot is the overall correlation of the plot.

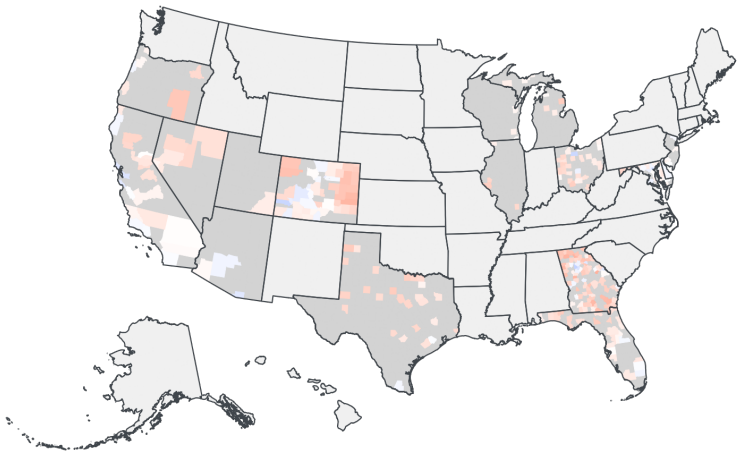

Figure B.3: Underlying Ideal Point Estimates in U.S. Counties. The plot displays the mean ideal point among voters in each county in the U.S. where I have data, mirroring Figure B.1. Counties that are more blue lean more towards the Democrats, whereas counties that are more red lean towards the Republicans. Otherwise, the figure is constructed in the same way as Figure B.1.

Table C.3: Multivariate regression of the VAE-vs-MCMC correlation on the tuned hyperparameters. Predictors are z-standardized, so each coefficient is the change in correlation associated with a one-standard-deviation increase in that hyperparameter; standard errors are in parentheses. * p<0.05, ** p<0.01, *** p<0.001.

	Coefficient
(Intercept)	0.856***
	(0.019)
Batch size	0.013
	(0.021)
Hidden size	0.011
	(0.020)
Embedding dim	-0.027
	(0.020)
Learning rate	-0.004
	(0.020)
ELBO samples	0.016
	(0.020)

linear regression, where the outcome is the correlation between the VAE and MCMC estimates, with the hyperparameters set as covariates and standardized so their coefficients are directly comparable (Table C.3). The results confirm the visual depiction – none of the hyperparameters has a coefficient that is statistically distinguishable from zero, so none of them substantively change the overall correlation when allowed to vary. I recommend researchers conduct a similar hyperparameter tuning process to ensure the best performance of the model for their specific application. The speed of the VAE model allows this kind of tuning process to be feasible even with large datasets, which is a significant advantage over traditional MCMC methods.

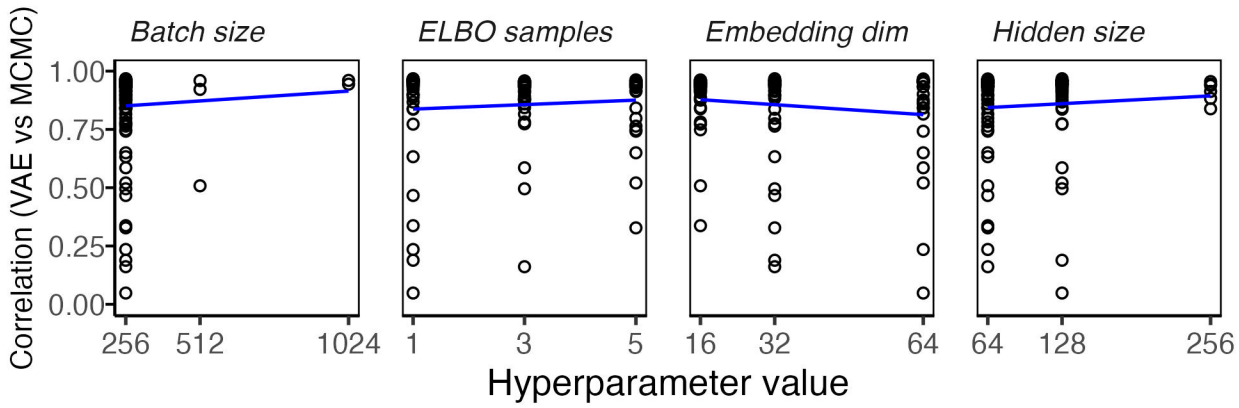

Figure C.4: Hyperparameter Tuning. The plot displays, for each iteration of the hyperparameter tuning process, the correlation between the VAE estimates and the MCMC estimates on the y-axis and the value of the hyperparameter on the x-axis. Each facet represents a different hyperparameter. The dashed black line is a loess line fit to the data.