Let the Experts Speak: Improving Survival Prediction & Calibration via Mixture-of-Experts Heads

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep mixture-of-experts models have attracted a lot of attention for survival analysis problems, particularly for their ability to cluster similar patients together. In practice, grouping often comes at the expense of key metrics such calibration error and predictive accuracy. This is due to the restrictive inductive bias that mixture-of-experts imposes, that predictions for individual patients must look like predictions for the group they're assigned to. Might we be able to discover patient group structure, where it exists, while *improving* calibration and predictive accuracy? In this work, we introduce several novel deep mixture-of-experts (MoE) based architectures for survival analysis problems, one of which achieves all desiderata: clustering, calibration, and predictive accuracy. We show that a key differentiator between this array of MoEs is how expressive their experts are. We find that more expressive experts that tailor predictions per patient outperform experts that rely on fixed group prototypes.

1 Introduction

AI has the potential to have a profound impact on clinical decision support systems (CDSS) (Eltawil et al., 2023). In this work, we focus on what clinicians care about most: highly accurate models, where probabilities have intuitive meaning (i.e., calibration), and the ability to reason by analogy to similar patients. Our work addresses survival analysis problems, where the task is to predict when clinical events will occur (i.e., time-to-event regression) while contending with right censoring (i.e., not observing the event time for all patients).

Mixture-of-experts (MoE) models for medical survival analysis are defined by two key components: 1) a router that assigns patients to groups and 2) a set of experts that produce event distributions for each group. We develop three novel deep MoE based survival architectures with an eye toward the above desiderata. Importantly, these three architectures differ only in the degree to which their experts customize their predictions for individual patients, allowing us to isolate the effect this has on the quality of predictive accuracy and calibration, which has not been carefully investigated in the medical survival modeling setting. The first architecture, **static MoE**, uses several experts where each expert learns an associated event distribution, which is then static across all patients. The second architecture, **adjustable MoE**, again learns a prototypical event distribution per expert, but can be adjusted to the individual patient. The third architecture, **dynamic MoE**, uses experts that each form a custom event distribution for individual patients. Only the third architecture is able to cluster patients *and* outperform strong baseline models on calibration and absolute error. The contributions of our work are as follows:

1. we develop three novel deep mixture-of-experts (MoE) based survival architectures, one of which achieves excellent clustering, calibration error, and absolute error.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

2. we report that the expressiveness of the experts is a key differentiator between deep MoE-based survival analysis models, supported by a targeted set of experiments on these three architectures.

2 Methods

We now describe our three proposed deep mixture-of-experts (MoE) based survival architectures. All models are feedforward deep learning models that use information from a patient's record (e.g., demographic data, physiological data, etc.) to forecast when they are likely to have a clinical event. Raw patient records $\mathbf{x}_0^i \in \mathbb{R}^a$ for $i = 1 \dots n$ are composed of categorical indicators (e.g., gender, etc.), standardized continuous features (e.g., heart rate), and any other available features (e.g., text embeddings of patient records, etc.). We learn embeddings for categorical indicators, which results in a d-dimensional feature vector that is fed to the model. Let $\mathbf{x} = \mathbf{x}_{\ell-1}^i \in \mathbb{R}^h$ be shorthand for the penultimate hidden state representation of our feedforward model with ℓ layers for the i^{th} patient. We will now describe the ℓ^{th} layer (i.e., the final layer), which is the MoE layer in all architectures. All methods are trained using an MTLR-style (Yu et al., 2011; Fotso, 2018) loss function, which is described in the appendix.

Static Mixture-of-Experts. The static MoE architecture, which learns an event distribution per expert that is then static with respect to all patients, is composed of a learnable router $W \in \mathbb{R}^{n \times h}$ where n is the number of experts and a collection of learnable experts $M \in \mathbb{R}^{n \times m}$, where each row can be mapped to a distribution over the m possible discrete event times. We produce a probability mass function (PMF) $\mathbf{p}$ over discrete event times as follows

$$\boldsymbol{\alpha}(\mathbf{x}) = \text{softmax}(\mathbf{x}W^\intercal/\kappa) \tag{1}$$

$$\mathbf{p} = \boldsymbol{\alpha}M' = \sum_{j=1}^{n} \alpha_j M'_j \tag{2}$$

where α_j is the weight on expert j, M'_j is the j^{th} row of the parameter matrix M after it has been normalized to form a discrete event distribution, and κ is a learnable temperature parameter that modulates the sharpness of expert selection. This architecture is the discrete-time analog to Hou et al.'s (2023) Deep Clustering Survival Machine.

Adjustable Mixture-of-Experts. The adjustable MoE learns an event distribution per expert and then *warps* it per patient. Let $\mathbf{t} \in [0,1]^m$ denote the canonical grid over the m discrete time bins, with $t_j = j/(m-1)$ for $j = 0, \dots, m-1$. Each expert k maintains a prototype vector $M_k \in \mathbb{R}^m$ of unnormalized scores over event times. To tailor expert k to patient i, we define a strictly monotone bijection between the expert's internal time $\boldsymbol{\tau} \in [0,1]^m$ and the canonical grid $\mathbf{t}$:

$$\underbrace{\phi_{k,\mathbf{x}} : \boldsymbol{\tau} \to \mathbf{t}}_{\text{forward}}, \qquad \underbrace{\psi_{k,\mathbf{x}} : \mathbf{t} \to \boldsymbol{\tau}}_{\text{inverse}} = \phi_{k,\mathbf{x}}^{-1}.$$

Concretely, we take the forward map to be a normalized mixture of two logistic CDFs,

$$F_{k,\mathbf{x}}(u) \;=\; \sum_{r=1}^{2} w_{k,r}(\mathbf{x})\, \sigma\big(a_{k,r}(\mathbf{x})\,[\,u - c_{k,r}(\mathbf{x})\,]\big), \quad \sigma(z) = \frac{1}{1+e^{-z}}, \tag{3}$$

with weights $w_{k,r}(\mathbf{x}) > 0$ and $\sum_r w_{k,r}(\mathbf{x}) = 1$, slopes $a_{k,r}(\mathbf{x}) > 0$, and ordered centers $0 < c_{k,1}(\mathbf{x}) < c_{k,2}(\mathbf{x}) < 1$. We enforce a mapping from $[0,1] \to [0,1]$ via endpoint normalization,

$$\tilde{F}_{k,\mathbf{x}}(u) \;=\; \frac{F_{k,\mathbf{x}}(u) - F_{k,\mathbf{x}}(0)}{F_{k,\mathbf{x}}(1) - F_{k,\mathbf{x}}(0)} \in [0,1], \tag{4}$$

and define $\phi_{k,\mathbf{x}}(u) = \tilde{F}_{k,\mathbf{x}}(u)$ and $\psi_{k,\mathbf{x}} = \phi_{k,\mathbf{x}}^{-1}$. Given a canonical gridpoint t_j, we compute $u_j = (m-1)\,\psi_{k,\mathbf{x}}(t_j)$ and linearly interpolate the prototype scores:

$$i_0 = \lfloor u_j \rfloor, \quad i_1 = \min(i_0 + 1, m-1), \quad w_j = u_j - i_0, \tag{5}$$

$$\tilde{M}_{k,j} \;=\; (1 - w_j)\, M_{k,i_0} + w_j\, M_{k,i_1}. \tag{6}$$

Let $\tilde{M}'_k$ denote the row-wise normalization of $\tilde{M}_k$ into an event-time PMF. The final per-patient PMF is then

$$\boldsymbol{\alpha}(\mathbf{x}) = \mathrm{softmax}\big(\mathbf{x}W^\intercal/\kappa\big), \tag{7}$$

$$\mathbf{p} = \boldsymbol{\alpha}\tilde{M}' = \sum_{k=1}^{n} \alpha_k(\mathbf{x})\, \tilde{M}'_k. \tag{8}$$

In practice, we evaluate $\psi_{k,\mathbf{x}}(t_j)$ via a batched bisection solver (see appendix for more details). This transformation family generalizes simple shift/scale warps and can emulate proportional-hazards style tilts while allowing richer early/late adjustments with minimal additional parameters per expert (Zhong et al., 2021; Nagpal et al., 2021).

Dynamic Mixture-of-Experts. The dynamic MoE architecture provides the most flexibility to the experts to form custom event distributions per patient. Since this architecture generates new expert distributions for each patient, we first project the final hidden state representaion $\mathbf{x}$ to a router representation with $\mathbf{x}_r = \mathbf{x}W_r^\intercal$ for $W_r \in \mathbb{R}^{h\times h}$ and an expert representation with $\mathbf{x}_e = \mathbf{x}W_e^\intercal$ for $W_e \in \mathbb{R}^{h\times h}$. We then divide the expert representation into n evenly-sized chunks $\mathbf{x}_{e,k}$ for $k = 1, \ldots, n$, which are each fed to a linear layer denoted $L_k \in \mathbb{R}^{m\times(h/n)}$ to form an event-distribution to obtain $M_k(\mathbf{x}_{e,k}) = \mathbf{x}_{e,k}L_k^\intercal$, which collectively form the dynamic matrix of unnormalized densities over event times $M(\mathbf{x}_e) \in \mathbb{R}^{n\times m}$. The final PMF is then

$$\boldsymbol{\alpha}(\mathbf{x}_r) = \mathrm{softmax}(\mathbf{x}_r W^\intercal/\kappa) \tag{9}$$

$$\mathbf{p} = \boldsymbol{\alpha}M(\mathbf{x}_e)' = \sum_{j=1}^{n} \alpha_j M(\mathbf{x}_e)'_j. \tag{10}$$

3 Experiments

We experiment with 3 datasets to probe the properties and capabilities of our proposed methods. SurvivalMNIST is a synthetic dataset (see Figure 1), which allows us to probe the models' abilities to predict event times and recover latent groups. We censor 15% of examples. SUPPORT2 is a survival analysis dataset with 9,105 examples and ∼32% censoring. Sepsis is a larger dataset with 40,336 patient records and only 2,932 positive instances of sepsis, making this a very challenging anomaly detection task (Reyna et al., 2020). The first 100 hours of each patient's ICU stay was summarized to perform a retrospective prediction, helpful when using partially labeled historical datasets to accelerate human labeling. Patients were administratively censored after 100 hours. We measure equal mass expected calibration error (ECE) (Roelofs et al., 2022) adjusted with inverse probability of censoring weighting (IPCW) and average over all time bins, concordance index, and absolute error adjusted with pseudo-observations (Qi et al., 2023). We also report the test set negative log-likelihood loss and parameter count for each method. All measurements are averaged over 5 random seeds. In order to obtain a ranking of the models by performance, for each random seed we take the difference between the MoE model's metric and MTLR model's metric, and then average that over the 5 runs to obtain an average performance gap. Network architectures and hyperparameters are described in the appendix.

4 Results

We report our results in Table 1. The SurvivalMNIST dataset represents the Platonic ideal of a dataset with clear latent groups and as a result the static MoE model performs best across all metrics, followed by dynamic MoE. This is a setting where the static MoE is perfectly specified, with exactly 10 expert heads, one for each digit. The only information needed to make Bayes-optimal predictions is to identify the digit, at which point the appropriate expert can predict the group's event distribution. Any attempt to adjust or customize the expert distributions per patient only adds unnecessary complexity and hurts performance. Nevertheless, we can see in Figure 1 that the dynamic MoE is able to recover the latent groups, with each expert specializing in a particular digit. The SurvivalMNIST dataset provides an interesting contrast to real-world datasets, where latent groups are almost never as well-defined.

Table 1: We report averages over 5 random seeds and in parentheses average deviation, given a random seed, from the MTLR baseline. Best results per dataset are bolded. ↓ indicates lower is better and ↑ indicates higher is better.

Dataset	Model	ECE ↓	Concordance ↑	Abs. Error ↓	Loss ↓	Parameters ↓
Survival MNIST	Static MoE	**0.004 (-0.003)**	**93.24 (0.69)**	**2.74 (-0.10)**	**2.197 (-0.041)**	209,847
	Adjustable MoE	0.008 (0.001)	92.48 (-0.08)	2.89 (0.04)	2.292 (0.054)	194,883
	Dynamic MoE	0.005 (-0.002)	92.65 (0.09)	2.81 (-0.03)	2.222 (-0.015)	195,731
	MTLR	0.006 (0.000)	92.56 (0.00)	2.84 (0.00)	2.238 (0.000)	187,189
SUPPORT2	Static MoE	0.066 (0.007)	79.68 (-0.48)	640.63 (3.12)	2.320 (0.131)	69,483
	Adjustable MoE	0.051 (-0.008)	79.65 (-0.52)	621.35 (-16.16)	2.322 (0.134)	69,435
	Dynamic MoE	**0.043 (-0.016)**	**80.97 (0.81)**	**619.89 (-17.62)**	**2.153 (-0.036)**	62,013
	MTLR	0.059 (0.000)	80.17 (0.00)	637.51 (0.00)	2.189 (0.000)	68,521
Sepsis	Static MoE	0.012 (0.002)	79.87 (-3.23)	4.37 (-0.04)	0.449 (0.032)	62,179
	Adjustable MoE	0.010 (-0.000)	81.48 (-1.62)	4.34 (-0.07)	0.433 (0.016)	62,835
	Dynamic MoE	**0.007 (-0.003)**	82.89 (-0.21)	**4.32 (-0.09)**	**0.412 (-0.005)**	57,269
	MTLR	0.010 (0.000)	**83.10 (0.00)**	4.41 (0.00)	0.418 (0.000)	62,241

Results on SUPPORT2 show the dynamic MoE model outperforms all other methods across all metrics. Importantly, the dynamic MoE model is outperforming the MTLR model on calibration error, concordance, absolute error, and loss, while the static and adjustable MoE models are, in general, not. We see similar results on the Sepsis dataset, providing further evidence that the dynamic MoE model is able to deliver on all desiderata: clustering, calibration, and predictive accuracy. Consistent with prior work that is subsumed by our adjustable MoE (e.g., Deep Cox Mixtures (Nagpal et al., 2021)), we observe that per-patient adjustments chiefly benefit calibration and predictive accuracy as measured by absolute error relative to MTLR, though we see limited impact on concordance or likelihood. Our dynamic MoE model extends those gains, delivering simultaneous improvements in calibration, accuracy, and discrimination, achieved with simple end-to-end learning rather than more elaborate training pipelines (e.g., spline estimation of baseline hazard rates, EM-based cluster estimation, etc.). The dynamic MoE model's ability to form custom event distributions per patient appears to be key to its strong performance on real-world datasets, where latent groups are not as well-defined as in SurvivalMNIST.

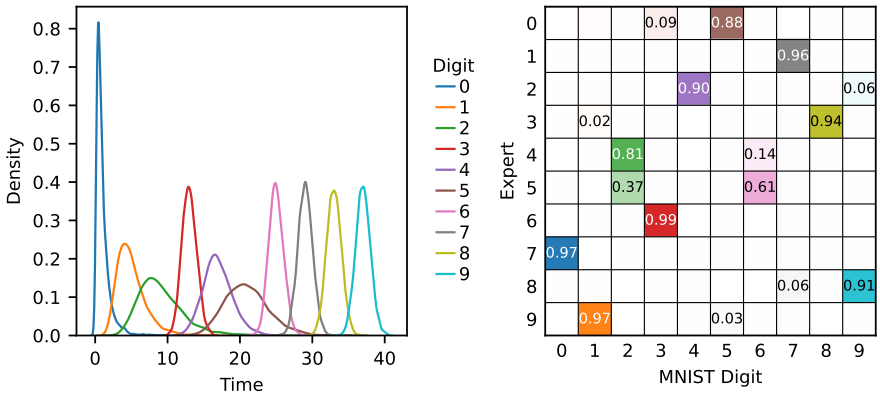

Figure 1: The left plot shows synthetic event distributions for each MNIST digit in SurvivalMNIST and the right plot shows which digits get routed to each expert in a **dynamic MoE** model.

5 Conclusion

We developed three novel deep mixture-of-experts (MoE) based survival architectures, one of which achieves excellent clustering, calibration error, and absolute error. We have shown that the expressiveness of the experts is a key differentiator between deep MoE-based survival analysis models, supported by a targeted set of experiments on these three architectures. Future work will explore the optimal allocation of parameters between the network backbone and the MoE layer, a wider range of adjustable MoE's given the rich space of possible parameter-efficient transformations of expert distributions, and how patients are routed to experts on real-world datasets.

References

Eltawil, F. A., Atalla, M., Boulos, E., Amirabadi, A., and Tyrrell, P. N. Analyzing Barriers and Enablers for the Acceptance of Artificial Intelligence Innovations into Radiology Practice: A Scoping Review. *Tomography*, 9(4):1443–1455, August 2023. ISSN 2379-139X. doi: 10.3390/tomography9040115.

Fotso, S. Deep Neural Networks for Survival Analysis Based on a Multi-Task Framework, January 2018.

Hou, B., Li, H., Jiao, Z., Zhou, Z., Zheng, H., and Fan, Y. Deep Clustering Survival Machines with Interpretable Expert Distributions. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pp. 1–4, April 2023. doi: 10.1109/ISBI53787.2023.10230844.

Nagpal, C., Yadlowsky, S., Rostamzadeh, N., and Heller, K. Deep Cox Mixtures for Survival Regression. In *Proceedings of the 6th Machine Learning for Healthcare Conference*, pp. 674–708. PMLR, October 2021.

Qi, S.-a., Kumar, N., Farrokh, M., Sun, W., Kuan, L.-H., Ranganath, R., Henao, R., and Greiner, R. An effective meaningful way to evaluate survival models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *ICML'23*, pp. 28244–28276, Honolulu, Hawaii, USA, July 2023. JMLR.org.

Reyna, M. A., Josef, C. S., Jeter, R., Shashikumar, S. P., Westover, M. B., Nemati, S., Clifford, G. D., and Sharma, A. Early Prediction of Sepsis From Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019. *Critical Care Medicine*, 48(2):210–217, February 2020. ISSN 0090-3493. doi: 10.1097/CCM.0000000000004145.

Roelofs, R., Cain, N., Shlens, J., and Mozer, M. C. Mitigating Bias in Calibration Error Estimation. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pp. 4036–4054. PMLR, May 2022.

Yu, C.-N., Greiner, R., Lin, H.-C., and Baracos, V. Learning Patient-Specific Cancer Survival Distributions as a Sequence of Dependent Regressors. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

Zhong, Q., Mueller, J. W., and Wang, J.-L. Deep Extended Hazard Models for Survival Analysis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 15111–15124. Curran Associates, Inc., 2021.

A Loss Functions

A.1 Uncensored Loss

All of our models are optimized using the Multitask Logistic Regression Loss (MTLR) (Yu et al., 2011; Fotso, 2018). We encode event times such that if patient i has a medical event at time s then all times $t \geq s$ will have label $y_t^i = 1$. Correspondingly, times $t < s$ will be labeled with 0. Typical label strings will look like a sequence of 0s followed by a sequence of 1s (e.g., $(0, 0, 0, 1, 1)$). The probability that our model assigns to the ground truth label sequence for a particular patient (the likelihood function) is then given as

$$p(\mathbf{Y} = (y_1, y_2, \ldots, y_m)|\mathbf{z}) = \frac{\exp(\sum_{j=1}^{m} y_j z_j)}{\sum_{k=0}^{m} \exp(f(\mathbf{z}, k))} \tag{11}$$

where $\mathbf{z} \in \mathbb{R}^m$ are logits from the model and $f(\mathbf{z}, k) = \sum_{j=0}^{k} 0 \cdot z_j + \sum_{j=k+1}^{m} 1 \cdot z_j$ for $0 \leq k \leq m$, which constrains the space of valid event sequences to runs of 0s followed by runs of 1s and corresponds to the disease occurring in the interval $[k, k+1)$. The boundary case is $f(\mathbf{z}, m) = 0$, which corresponds to the sequence of all 0s. We can minimize the negative log-likelihood for an uncensored patient with the following loss

$$\mathcal{L}_{\text{uncensored}} = -\sum_{j=1}^{m} y_j z_j - \log\left(\sum_{k=0}^{m} \exp(f(\mathbf{z}, k))\right) \tag{12}$$

We now provide more detail on how to exchange between the model's logits $\mathbf{z}$ and a PMF $\mathbf{p} \in [0, 1]^m$ over event times. Our model produces *increment logits* due to its interaction with the cumulative-sum parameterization of the softmax (Equation 11). However, all methods must blend distributions over event times from different experts in probability space and therefore we rely on Equation 11 to map from increment logit space to probability space. We must also define an inverse operation to map from a PMF back to increment logits. We have

$$z_j = \log(p_j) - \log(p_{j+1}) \text{ for } j = 1, \ldots, m-1 \tag{13}$$

and set $z_m = 0$. We now derive this inverse operation for completeness. Let

$$u_t = \sum_{j=t}^{m} z_j \text{ for } t = 1, \ldots, m. \tag{14}$$

By Equation 11, $p_j = \frac{\exp(u_j)}{\sum_{k=0}^{m} \exp(f(\mathbf{z},k))}$ for $j = 1, \ldots, m$, which is equivalent to $\text{softmax}(\mathbf{u})$ for $\mathbf{u} \in \mathbb{R}^m$. Recall that the softmax function is invariant to additive shifts (i.e., $\text{softmax}(\mathbf{u}) = \text{softmax}(\mathbf{u}+c)$ for any constant c). We can therefore set $c = -u_m$ so that $u_m = 0$ and by Equation 14, $z_m = 0$. We can now write $u_j = \log(p_j) + c'$ for $j = 1, \ldots, m$. Since $u_m = 0$, we have $c' = -\log(p_m)$ and therefore $u_j = \log(p_j) - \log(p_m)$ for $j = 1, \ldots, m$. By Equation 14, we have

$$\begin{aligned} z_j &= u_j - u_{j+1} && (15)\\ &= \log(p_j) - \log(p_m) - \log(p_{j+1}) + \log(p_m) && (16)\\ &= \log(p_j) - \log(p_{j+1}) \text{ for } j = 1, \ldots, m-1 && (17) \end{aligned}$$

and $z_m = 0$, which completes the derivation.

A.2 Censored Loss

We now describe how to contend with censored data, which occurs when we do not observe if or when a patient has an event. Suppose a patient is censored at time s_c and time $t + t_j$ is the closest time point after s_c. Then all sequences $\mathbf{Y} = (y_1, y_2, \ldots, y_m)$ with $y_i = 0$ for $i < j$ are consistent with this censored observation. Therefore the likelihood for a censored patient is the survival function (i.e., 1 minus the cumulative density function (CDF)), which is

$$p(S \geq t + t_j|\mathbf{z}) = \frac{\sum_{k=j}^{m} \exp(f(\mathbf{z}, k))}{\sum_{k=0}^{m} \exp(f(\mathbf{z}, k))} \tag{18}$$

and in turn, the negative log-likelihood loss for a censored patient is

$$\mathcal{L}_{\text{censored}} = -\left[\log\left(\sum_{k=j}^{m}\exp(f(\mathbf{z},k))\right) - \log\left(\sum_{k=0}^{m}\exp(f(\mathbf{z},k))\right)\right]. \tag{19}$$

A.3 Regularizers

We apply a load-balancing loss to all MoE models to ensure the model uses all available experts. Let $\bar{\boldsymbol{\alpha}} = \frac{1}{b}\sum_{i=1}^{b}\boldsymbol{\alpha}(\mathbf{x}^i)$ be the average expert distribution over a batch of size b. The load-balancing loss encourages the model to use all experts equally across the batch of examples by penalizing low-entropy average expert distributions. For a given batch, the loss is given as

$$\mathcal{L}_{\text{load-balance}} = \lambda_{\text{lb}} \cdot n \sum_{i=1}^{n} \bar{\alpha}_i^2, \tag{20}$$

where n is the number of experts and λ_{lb} is a hyperparameter that controls the strength of the regularization. This loss is minimized when $\bar{\boldsymbol{\alpha}}$ is the uniform distribution.

B Hyperparameters and Training Details

Table 2: Model specific hyperparameters for SUPPORT2 and Sepsis datasets.

Model	Static MoE	Adjustable MoE	Dynamic MoE	MTLR
Hidden Dim. (h)	176	186	128	176
Num. Layers (ℓ)	2	2	1	2
Num. Experts (n)	10	10	8	-

Table 3: Model specific hyperparameters for SurvivalMNIST.

Model	Static MoE	Adjustable MoE	Dynamic MoE	MTLR
Hidden Dim. (h)	208	186	160	176
Num. Layers (ℓ)	2	2	1	2
Num. Experts (n)	10	10	10	-

Table 4: Shared hyperparameters for all models.

Hyperparameter	Value
Learning Rate	1e-3
Batch Size	64
λ_{lb}	0.01
κ Init.	2.0
m	100

We define a validation set for SurvivalMNIST by randomly sampling 5,000 examples from the training set and then use the provided test set for final evaluation. For SUPPORT2 and Sepsis, we randomly sample 10% of all examples to form a validation set and sample another 10% to form a test set. We use the validation set to perform minimal hyperparameter tuning and in general defaulted to the same hyperparameters across all models with the exception of parameter counts to ensure a fair comparison. We train all models to convergence as measured by the validation set loss, using early stopping with a patience of 10 epochs. All models are trained with the Adam optimizer. We use a hidden dimension of 128-208 and 1-2 hidden layers for all models, which are fully connected layers with ReLU activations. We use 8-10 experts for all MoE models. The number of discrete time bins is set to $m = 100$ for all datasets. The load-balancing loss weight is set to $\lambda_{\text{lb}} = 0.01$ for all MoE

models. The temperature parameter κ is initialized to 2.0 for all MoE models and learned during training. All models are implemented in PyTorch and trained on a single GPU. Our Github code repository will be released upon publication.

C Inversion and Gradients for the Two-Logistic Warp

A. Inversion by Bisection

Recall the patient- and expert-specific forward map

$$F_{k,\mathbf{x}}(u) \;=\; \sum_{r=1}^{2} w_{k,r}(\mathbf{x})\,\sigma\big(a_{k,r}(\mathbf{x})[u - c_{k,r}(\mathbf{x})]\big), \qquad \sigma(z) = \tfrac{1}{1+e^{-z}}, \tag{21}$$

and its endpoint-normalized version

$$\tilde{F}_{k,\mathbf{x}}(u) \;=\; \frac{F_{k,\mathbf{x}}(u) - F_{k,\mathbf{x}}(0)}{F_{k,\mathbf{x}}(1) - F_{k,\mathbf{x}}(0)} \in [0,1]. \tag{22}$$

We define

$$\phi_{k,\mathbf{x}}(u) = \tilde{F}_{k,\mathbf{x}}(u) \quad \text{and} \quad \psi_{k,\mathbf{x}} = \phi_{k,\mathbf{x}}^{-1}.$$

Since σ is strictly increasing and $w_{k,r}(\mathbf{x}), a_{k,r}(\mathbf{x}) > 0$ with $0 < c_{k,1}(\mathbf{x}) < c_{k,2}(\mathbf{x}) < 1$, $F_{k,\mathbf{x}}$ (and hence $\tilde{F}_{k,\mathbf{x}}$) is strictly increasing on $[0,1]$, so the inverse exists and is unique.

For a given $(k, \mathbf{x}, t_j)$ we obtain $\tau^\star = \psi_{k,\mathbf{x}}(t_j)$ as the unique solution to

$$g(\tau;\theta) \;=\; \tilde{F}_{k,\mathbf{x}}(\tau;\theta) - t_j \;=\; 0, \quad \theta = (w_{k,1:2}, a_{k,1:2}, c_{k,1:2}).$$

Because $g(0) \le 0 \le g(1)$ and g is strictly increasing, *bisection* converges to $\tau^\star$ with bracketing on $[0,1]$.

Vectorized bisection (batched). For all items in a batch and all experts/time-bins (indices suppressed):

1. Initialize lo $\leftarrow 0$, hi $\leftarrow 1$.
2. For $s = 1, \ldots, S = 20$:
   (a) mid $\leftarrow$ (lo + hi)/2.
   (b) $v \leftarrow \tilde{F}_{k,\mathbf{x}}(\text{mid}) - t_j$.
   (c) Update lo $\leftarrow \mathbf{1}_{\{v<0\}} \cdot$ mid $+ \mathbf{1}_{\{v\ge 0\}} \cdot$ lo, hi $\leftarrow \mathbf{1}_{\{v<0\}} \cdot$ hi $+ \mathbf{1}_{\{v\ge 0\}} \cdot$ mid.
3. Return $\tau^\star \approx$ (lo + hi)/2.

This procedure halves the bracketing interval at each step, so $S = 20$ iterations yield $\approx 10^{-6}$ precision.

Numerical safeguards. We clip the denominator $D = F_{k,\mathbf{x}}(1) - F_{k,\mathbf{x}}(0)$ away from 0 when forming $\tilde{F}_{k,\mathbf{x}}$, bound the slopes $a_{k,r}(\mathbf{x}) \in [a_{\min}, a_{\max}]$ (e.g., $a_{\min} = 0.1$, $a_{\max} = 35$), enforce $w_{k,1:2}$ via a softmax, and enforce ordered centers by a stick-breaking parameterization to improve conditioning.

B. Gradients via the Implicit Function Theorem

Let $\tau^\star = \psi_{k,\mathbf{x}}(t_j)$ satisfy $g(\tau^\star;\theta) = 0$ with $g(\tau;\theta) = \tilde{F}_{k,\mathbf{x}}(\tau;\theta) - t_j$. Because $\partial_\tau \tilde{F}_{k,\mathbf{x}}(\tau^\star;\theta) > 0$, the implicit function theorem gives

$$\frac{\partial \tau^\star}{\partial \theta} \;=\; -\,\frac{\partial_\theta \tilde{F}_{k,\mathbf{x}}(\tau^\star;\theta)}{\partial_\tau \tilde{F}_{k,\mathbf{x}}(\tau^\star;\theta)}. \tag{23}$$

Write $F = F_{k,\mathbf{x}}$, $F_0 = F(0)$, $F_1 = F(1)$, and $D = F_1 - F_0$. Using

$$\tilde{F}(\tau) = \frac{F(\tau) - F_0}{D},$$

we obtain

$$\partial_\tau \tilde{F}(\tau) = \frac{\partial_\tau F(\tau)}{D}, \qquad \partial_\theta \tilde{F}(\tau) = \frac{\partial_\theta F(\tau) \;-\; \partial_\theta F_0 \;-\; \tilde{F}(\tau)\,\big(\partial_\theta F_1 - \partial_\theta F_0\big)}{D}. \tag{24}$$

For $F(\tau) = \sum_{r=1}^{2} w_r\, \sigma\big(a_r(\tau - c_r)\big)$ we have the elementary partials

$$\partial_\tau F(\tau) = \sum_{r=1}^{2} w_r\, a_r\, \sigma'\big(a_r(\tau - c_r)\big), \qquad \sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big), \tag{25}$$

$$\partial_{w_r} F(\tau) = \sigma\big(a_r(\tau - c_r)\big), \qquad \partial_{a_r} F(\tau) = w_r\,(\tau - c_r)\,\sigma'\big(a_r(\tau - c_r)\big), \tag{26}$$

$$\partial_{c_r} F(\tau) = -\,w_r\, a_r\, \sigma'\big(a_r(\tau - c_r)\big), \tag{27}$$

and the same forms evaluated at $\tau = 0$ and $\tau = 1$ for F_0 and F_1. Substituting (24) into (23) yields closed-form expressions for $\partial\tau^\star/\partial\theta$.